Cloud Computing Made Easy

S.SRIDHAR
Professor & Director
RV Centre for Cognitive & Central Computing
R.V.College of Engineering, Mysore Road
Bangalore-560059 India

Abstract: **Cloud computing is the delivery of computing as a service rather than as a product, where by shared resources software and information are provided to computer are other device as a utility over a network. In a cloud computing system, there is a significant workload shift. Local computers no longer have to do all the heavy lifting, when it comes to running applications. The network of computers that makeup the cloud handles them instead. Hardware and software demand on the users side decrease. The only thing the users' computer needs to be able to run is the cloud computing systems interface software, which can be as simple as a web browser and the clouds network take care of the rest. This article is prepared based on the Author's teaching the subject for M.Tech level recent years, keeping in view of VTU Syllabus in particular.**

Key words: **Cloud, delivery of computing, hardware and software demand, interface software, web browser, cloud network**

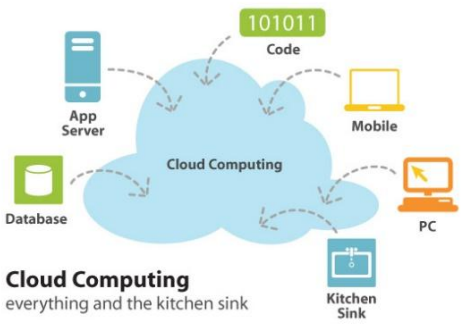
INTRODUCTION

Computing enables provisioning of standardized business and computing service through shared infrastructure. It reduces the cost by providing best IT service.It helps with dynamic infrastructure and helps by providing good cloud delivery model. It helps organizing in cloud strategy, design and development of cloud road map and helps IT benefit on Return of investment [ROI]. Cloud computing helps client utilization the IT resource to display new applications, services or computing resources rapidly without regenerating their entire infrastructure in dynamic manner. Efficiency is achieved by virtualization, organization gets benefited by several storage automation usage and standardization through service ordering, flexible pricing. The cloud model may be as follows:-

Public Cloud: Where the business rents the capacity and then pays for what they use ;

Private cloud: The business turns their IT environment into a cloud, uses it to deliver services to their users;

Hybrid Cloud: Includes both, wherein, the public cloud computing is accessed through internet while Private cloud computing exists behind the firewall.



Cloud and Dynamic Infrastructure

Cloud computing helps client utilizing the IT resources to display new application, services or computing resources rapidly without re-engineering their entire infrastructure in dynamic manner. Cloud dynamic infrastructure is based on architecture that combines :- Service Management. This deals with management of services [infrastructure] dynamically and efficiently monitoring it; Asset Management : Maximize the value of critical business and IT assets; Virtualization and Consolidation: This deals with reducing the operating cost; Information Infrastructure :Helps business improvement with security and availability issue.

a) Energy efficiency: Address energy and sustainability challenges across business and IT infrastructure.

b) Security: Provides end to end governance and risk management for business.

c) Resilience: Maintains continuous business and IT operations while responding to risk.

Cloud Computing Delivery Models and Services

Service delivery in cloud computing companies with three different service models:

i. ***Software – as – a – service (SaaS):-*** provides complete application to a clouds end user. It is mainly accessed through a web portal and service oriented architecture based on web service technologies. Eg. Google drive.

ii. ***Platform– as– a– Service (PaaS):-***Comprises the environment for developing and provisioning cloud applications. The main user of this layer are developers. The services offered on a cloud platform tend to represent a compromise between complexity and flexibility that allows application to be implemented quickly and loaded in the cloud without much configuration.

iii. ***Infrastructure– as– a– Service (IaaS):-***This provides essential IT resources like data storage resources, computing resources and communication channel. Physical resources are abstracted by virtualization, which means they can be shared by several OS and end user environments on the virtual resources ideally without any mutual interference. These virtualized resources usually comprise CPU and RAM, data storage resources.

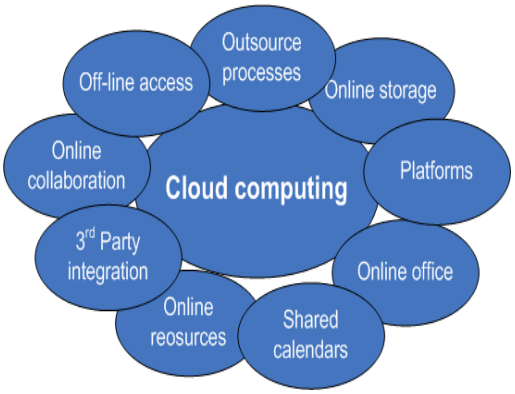
iv. ***Ethical Issues:-*** Cloud computing is based on a paradigm shift with profound implications for computing ethics. The main elements of this shift are:-

   a. The control is relinquished to third party services.

   b. The data is stored on multiple sites administrated by several organizations.

   c. Multiple services interoperate across the network.

Unauthorized access, data corruption, infrastructure failure and service unavailability are some of the risks related to relinquishing the control to third party service. Moreover whenever a problem occurs it is difficult to identify the sources and entity causing it. Unlimited data sharing and storage among organizations test the self-determination of information, the right or ability of individuals to exercise personal control over the collection and use and disclosure of their personal data by others, this test the confidence and trust in todays evolving information.



v. Cloud Vulnerabilities

Clouds are affected by malicious attacks and failures of the infrastructure (eg; power failure). Such events can affect Internet domain name servers and prevent access to clouds or can directly affect clouds. For example, an attack at Akamai Technologies on June 15, 2004, caused a domain name outage and a major blackout that affected Google Inc., Yahoo! Inc. and many other sites. In May 2009, Google was the target of a serious denial-of-service (DoS) attack that took down services such as Google News and Gmail for several days. Lightning caused a prolonged downtime at Amazon.com Inc. on June 29 and 30, 2012.

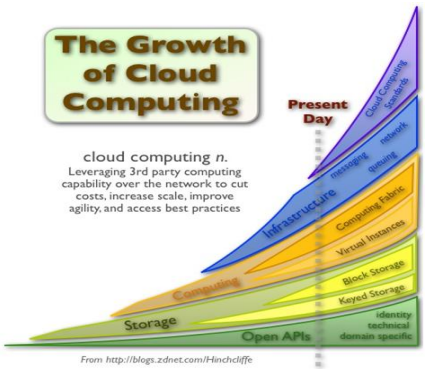
Details on Cloud Computing Characteristics and Cloud Adopting

Cloud computing provides commodity based hardware and software where in the service is able to be moved from one cloud provider to another without affecting the service. This provides "PAY-AS-YOU-GO" service. Four major barriers for large scale adoption of cloud service are as follows:-

- Security: Limited knowledge of the physical location of data adoption of virtualization concept.
- Governance and regulatory compliance: Data governance model for cloud services, and ensuring data privacy
- Integration and interoperability: Taking DecisionWhether to display or not to display the workload on the cloud.

Cloud Adoption: The applications that are suitable for cloud adoption are: Low priority business

application; Low availability requirement and short life span applications. Based on technical issues, the applications that are suitable for cloud adoption are: Those which are modular and loosely erupted; Software developing and testing. Cloud adaption is well suited for those applications that need different level of infrastructure throughout the day. Cloud adoption is not suited for applications that are mission critical. ie, for data not suitable for those applications that require high level auditability, accounting, high bandwidth and so on.



Cloud Computing at Amazon

Amazon introduced a computing platform that has changed the face of computing in the last decade. First it installed a powerful computing infrastructure to sustain its core business, e-commerce, selling a variety of goods ranging from books and CDs to gourmetfoods and home appliances. Then Amazon discovered that this infrastructure can be further extended to provide affordable and easy to use resources for enterprise computing, as well as, computing for the masses.

	EGEE Grid	Amazon Cloud
Target Group	Scientific community	Business
Service	short-lived batch-style processing (job execution)	long-lived services based on hardware virtualization
SLA	Local (between the EGEE project and the resource providers)	Global (between Amazon and users)
User Interface	High-level interfaces	HTTP(S), REST, SOAP, Java API, BitTorrent
Resource-side middleware	Open Source (Apache 2.0)	Proprietary
Ease of Use	Heavy	Light
Ease of Deployment	Heavy	Unknown
Resource Management	probably similar	
Funding Model	Publicly funded	Commercial

Summary of „An EGEE Comparative Study: Grids and Clouds Evolution or Revolution" by Markus Klems

Measured Service, Cloud Models

Measured service: The reason why organizations wish to migrate to cloud computing is: Flexibility, cost reduction; To avoid vulnerability and delay in data delivery service.

Cost factor: They have the Service Level Agreement (SLA's. And in Labour perspective, the labour need not posses operating system skills, middleware skills, database skills and application skills. (Cloud computing does the needful).

Benefit of Cloud adoption areas: Self service capability; Resource availability; Operational efficiency. The cloud models are as follows: -

- Public Cloud: Here the business results the capability and then pays for what they use on demand. The word 'Public' refers to freely available in the sense, users data are publically available on pay per use basis.Applications suitable are web pages, wiki blog.
- Private Cloud: Business turns the IT environment into a cloud and uses it to deliver services to their users. Services that private cloud provides are virtualization, security. Access control and management by consuming services from public cloud.ie, Business provides cloud service to the users by consuming services from public cloud.
- Hybrid Cloud: It is the combination of interoperating public and private cloud. Private cloud computing exists behind the firewall where as public cloud is accessed through internet. This deals with: Community Cloud; Shared private Cloud; Dedicated private Cloud; Dynamic private Cloud.

Security in Public Cloud, Public Versus Private Clouds and Cloud Infrastructure Self Service

Multi tenancy: When the higher risk clients are provided with the security, the lower risk client gets higher security than higher risk clients when the clients use the same cloud, there is possibility that attacks made to one client may affect another.

Security assessment: The cloud provider should perform regular security assessments through person who is experienced to identify issues and fix them.

Shared Risk: The cloud services provide that are built for layering SaaS on top of IaaS and it can affect Cloud users Security.

Staff Security Servicing: The Contractors should go through full background investigation as compared to Regular Employee.

Distributed Data centers: Cloud computing environment should be less prone to disasters. This is achieved by geographical distribution of data centers. Those providers who are not geographically distributed have to undergo regular and disaster recovery test that includes SLA's

Physical security: Physical security threats should be analyzed before choosing a cloud security provider regarding biometric access, omit guards etc.

Policies: Cloud provides should include incident response policies.

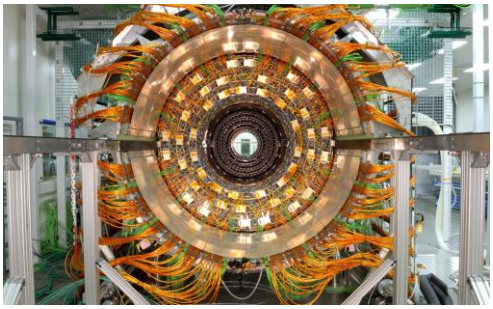
Coding: Should follow secure coding practices and written using standard methodology.

Data Leakage: This is the greatest organizational risk from security point of view. Hence data has to be encrypted while in flight at rest.

Public Cloud is the shared cloud infrastructure where in the business uses the cloud service and then pays for what they use on demand. Applications suitable are web pages, wiki blogs etc.

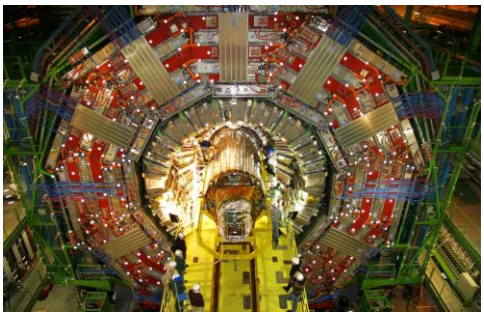
Private Clouds

Business turns their IT environment into a cloud and uses it to deliver services to their users. ie, business provide cloud service to the users by connecting services from the public cloud.



Cloud Infrastructure Self Service

Self service provisioning of infrastructure is possible in private cloud. Cost savings are possible from providers to large enterprise wide solution. Infrastructure strategy and planning for cloud computing help the client in providing assistance in understanding the business value that cloud computing brings. The path to cloud computing involves the following:- Server Virtualization, Distributed Virtualization, Public Cloud, Private Cloud and Hybrid Cloud.



Gamut of Cloud Solutions, Principal Technologies

There are 5 Categories:-

a) PaaS

Platform as a service provides the foundation to build highly scalable a robust web base application in the way similar as it would be done in Windows & Linux. It saves cost by reducing Software Licensing, Infrastructure Cost. Cost of developing, testing and hosting environment. Eg: Google App Engine, Heroku etc.

b) SaaS

This provides the computing hardware, software as well as the solution.It saves cost by Reducing software license a infrastructure cost; Reduces support, maintenance and administration cost; Reduce cost of development, delivery and maintenance.Eg. Google Apps, Microsoft 365.

c) IaaS

This service is used by software administation. This service is to provide infrastructure for the application.It saves cost byEliminating the need to our provision resource, Reduce capital expenditures on infrastructure.It requires dynamic scale up and down of resources.

d) Storage as a Service

This is the provisioning of DB like service billed on the utility computing basis. Eg: GB/Month.

e) Desktop as a Service

This is the provisioning of the desktop environment either within or within a browser of terminal server.

Principal Technologies The main drives for cloud computing are cost, agility and time to market.; The increased pooling and sharing of resources increases cost saving in IT infrastructure, agility and faster time to market for application; Virtualization is the foundation for the cloud.

Cloud Strategy and Cloud Design

The key steps in cloud implementation planning are: Understanding cloud strategy; Define cloud application requirement; Access cloud readiness; Define high level cloud architecture; Identifying change management requirements; Develop roadmap and implementation plan.

Cloud deployment results in following benefits:- Reduce Risk and faster development; Improved Service to achieve IT objective; Lower infrastructure cost.

Cloud Design

Service oriented Architecture (SOA) is very useful architecture for implementing applications in cloud. Standardizing across the different cloud environment is not possible because of the heterogeneity in nature. The cloud based services consist of a coherent set of business processes that are aligned to the business boundaries.

Architecture overview : The reason for architecture overview is to communicate to the spousor and external stake holders and developers; The artifact gives the high level view of the cloud architecture landscape; The architecture summary ensures / represents the governing ideas of cloud based offering and enterprise architecture; Provides overview of main offering and relationship with components, nodes, connectors, external system etc.

Amazon Web Services

Amazon was the first provider of cloud computing; it announced a limited public beta release of its Elastic Computing platform called EC2 inAugust 2006.

i. ***Elastic Compute Cloud (EC2)*** is a web service with a simple interface for launching instances of an application under several operating systems, such as several Linux distributions,Microsoft Windows Server 2003 and 2008, OpenSolaris, FreeBSD, and NetBSD.A user can interact with EC2 using a set of SOAP messages, and can list available AMI images, boot an instance from an image, terminate an image, display the running instances of a user, display console output, and so on.

ii. ***Simple Storage System (S3)*** is a storage service designed to store large objects. It supports a minimal set of functions: write, read, and delete.

iii. ***Elastic Block Store (EBS)*** provides persistent block level storage volumes for use with Amazon EC2 instances.

iv. ***Simple DB*** is a non-relational data store that allows developers to store and query data items via web services requests;

v. ***Simple Queue Service (SQS)*** is a hosted message queue.

vi. ***CloudWatch*** is a monitoring infrastructure used by application developers, users, and system administrators to collect and track metrics important for optimizing the performance of applications and for increasing the efficiency of resource utilization.

vii. ***Virtual Private Cloud*** etc

Today Amazon offers cloud services through a network of data centers on several continents.

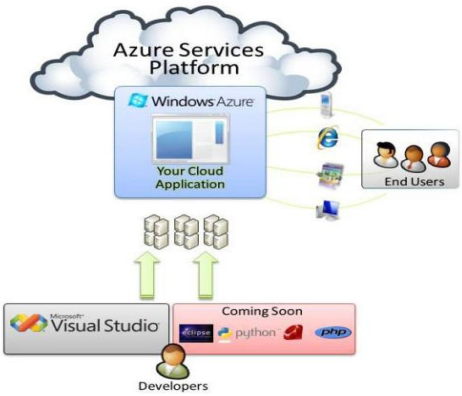
Amazon charges a fee for EC2 instances, EBS storage, data transfer, and several other services.

Cloud Computing, the Google Perspective

Google's effort is concentrated in the area of Software-as-a-Service (SaaS). It is estimated that the number of servers used by Google is close to 1.8 million in January 2012 and is expected to reach close to 2.4 million in the early. Services such as Gmail, Google Drive, Google Calendar, Picasa and Google Groups are free of charge for individual users and available for a fee for organizations. These services are running on a cloud and can be invoked from a broad spectrum of devices including mobile ones such as iPhones, iPads, Black Berrys, and laptops and tablets; the data for these services is stored at data centers on the cloud. The Gmail service hosts emails on Google servers, provides a web interface to access them and tools for migrating from Lotus Notes and Microsoft Exchange. Google docs is a web-based software for building text documents, spread sheets and presentations. Google Calendar is a browser-based scheduler; it supports multiple calendars for a user, the ability to share a calendar with other users, the display of daily/weekly/monthly views, to search events, and to synchronize with the Outlook Calendar. Google is also a leader in the Platform-as-a-Service (PaaS) space. AppEngine is a developer platform hosted on the cloud; initially it supported only Python and support for Java was added later; detailed documentation for Java is available.

Microsoft Windows Azure and Online Services

Azure and Online Services are PaaS (Platforms-as-a-Service) and, respectively, SaaS (Software as a Service) cloud platforms from Microsoft. Windows Azure is an operating system, SQL Azure is a cloud-based version of the SQL Server, and Azure AppFabric (formerly .NET Services) is a collection of services for cloud applications. Windows Azure has three core components : Compute which provides acomputation environment, Storage for scalable storage, and Fabric Controller which deploys, manages, and monitors applications; it interconnects nodes consisting of servers, high-speed connections, and switches. The Content Delivery Network (CDN) maintains cache copies of data to speed up computations. The Connect subsystem supports IP connections between the users and their applications running on Windows Azure. The API interface to Windows Azure is built on REST, HTTP and XML.



Implementation Using SOA

Service oriented architecture (SOA) is very useful architecture style for implementing applications in the cloud. This provides the best way to utilize the application services provided by the cloud. Standardizing across the different environment is not possible and hence they consist of heterogeneous environments. Language and platform independent services can be provided using standards band platform agnostic SOA architecture.

Conceptual Cloud Model

This includes the set of cloud band offerings that make up cloud based business solution. This provides abstract view of the design of a cloud application to business stakeholders. This model should contain following elements:- Conceptual structure of cloud application; Dynamic interaction and dependencies among various conceptual components; The components that comprise of cloud services provided by the application. This model should include:- Cloud application security and privacy principles; Governance through Authentication Access Control, Data protection, Logging and alerting.

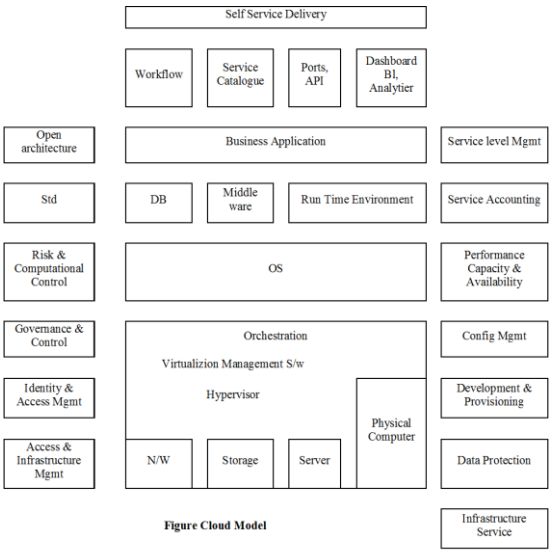


Figure Cloud Model

Cloud Service Defined

Service definition should include services delivered, services portfolio, service component, service owner, process, service level agreement and management and their objectives. Service scope overview consists of Platform integration and deployment services include Order management, Platform built test, Installation, Platform removal and return.

Software Platform Management Services

This contains: Software Platform Management services; Software Platform Design consulting; Software Platform support and maintenance; Software Platform creation and customization; Software Delivery; Antivirus Management; Service to user/customer initiated service requirement.

Cloud Ecosystem

Eco system focused on LaaS or PaaS cloud service.

Cloud Business Process Management (BPM)

Business process Management governs an organizations core business process. Cloud environment helps in following ways. Integration of core business; Value focused efficiency; Continuous improvement and Cultural considering of organ.

Identifying BPM Opportunities

The following questions may help one to identify cloud opportunity better : 1.How do you manage your core BP? 2. What are your current process initiatives? 3. What are your current process governance facilities? 4. What pdts your orgin have? What type of product? 5. Has your orgin adopted SOA?

Cloud Technical Strategy

How cloud customers can enable cloud deployment.This enable orgin, Built middleware cloud in data center, Utilize public clouds.

Cloud Enabled Middleware Services

Infrastructure services; Platform services; App services.

Cloud Use Cases

The cloud use cases include problem relation to the following use cases and the corresponding solutions:- Infrastructure as a service (IaaS) or Test / Development; Standardized development platform / middleware; Application cloud; SaaS to end customers; Cloud Service Management

Service Management System Provides

Visibility and Automation needed for efficient cloud delivery in Public & Private cloud.

This includes : Simplify user interaction with IT, Enable policies to lower cost with provisioning, Increase system administration productivity.

Three Categories of Opportunities for Cloud Broken

Cloud service intermediation; Cloud aggregation; Cloud service arbitrage.

Key Cloud Solution Characteristics

The solution characteristics for cloud services involves: Scalability, High availability, Application life cycle, Policies, Application awareness & policy based allocation, Resource awareness and Policy based allocation.

Cloud Stack, Computing On Demand, And Cloud Sourcing

Cloud computing in Demand.(COD): On demand computing is the need of the hour and is essential in supercomputing environment even. COD allows user to Align cost with utilization, Increase end users availability significantly, Balance workload dynamically across servers, React to short term resources requirement, Reduce the physical footprint, Increase system utilization, Double the workload delivered.This includes: - Pre-provisioning, On demand CPU / Memory / VM resources, Dynamic capacity [Processor], Dynamic capacity platform.

Cloud Analytics, Testing under Cloud, Information Security

It is the new offering in the era of cloud computing. It provides users with better forecasting technique to analyze and optimize the service lines and provides a higher level of accuracy. Cloud analytics can help them apply analytic principles and best practices to analyze the different business consequences and achieve newer levels of optimization. Analytics works with the combination of hardware service and middleware. This expertise makes it best suited to help clients extract new value from their business information.

Testing under cloud

Private cloud development need virtualized services that can be used for testing the resources for the cloud. This ensures more secure and scalable solutions where consumers can access the IT resources in the test environment.

Benefits

Cut capital and operation costs and not affected mission critical production applications; Offer new and innovative services to client and present and opportunity to speed cycles of innovation and improve solution quality; Facilitate a test environment based on request and provide request based service for service network and OS.

Information security

Information security risks are potential damage to information assets. By knowing the current status, a risk based approach can be taken. Risk can be qualified by expected (average) damage.

Value of asset

What are your valuable information assets?

Vulnerabilities

What a vulnerabilities exists in your system that can be exploited and lead the damage of assets ?

Threats

The level of threats that aim at exploiting vulnerability.Security control are safeguards or counter measures to avoid or minimize information security risks :

Must be effective

Mitigate the given risk;

Should be adaptive

Adopt to changing risks.

Virtual Desktop Infrastructure

Virtual desktop infrastructure provides end-user virtualization software. This is designed to help transform distributed IT architectures into virtualized, open standards based framework leveraging centralized IT service. VDI combines hardware, software and services to connect your clients authorized user to platform independent, centrally managed application and full desktop images as virtual machines on servers.

Storage Cloud

For any type of cloud deployment, whether a private, public or a hybrid cloud, the environments are built using key foundation building blocks, such as servers, storage, applications and infrastructure. Storage and compute resources scale together are failure to manage them efficiently as result of cloud service.Storage management in cloud can help organizations to address their challenges around data and storage management in their cloud – availability of data at all times, storage resource utilization, application performance.

Cloud management

Managing cloud virtualized infrastructure can present organizations significant challenges in areas related to implementation and service management. They are:-

Resilancy

Resilancy is the capacity to rapidly adopt and respond to risks as well as opportunities. This

maintains continuous business operations that support growth and operate in potentially adverse conditions:- From ***facilities perspective***, you may want to implement power protection; ***Security perspective:*** to protect application and data implement biometrics solution; ***Process perspective:*** implement identification and documentation of critical business process; ***Organizational perspective:*** Address to geographical diversity, backup of workstation date.

Provisioning

Provisioning process is a service that uses a group of compliant process called "Solution Realization". Environment provisioning roles separate proportion tasks and accurate tasks from provisioning tasks: - Provisioned products are servers built with all the software and infrastructure required to support a business application; Standard solution are defined so that standard work flow can be derived; Design is completed with due diligence before the request for service is accepted.

Asset Management

Asset management is an important port of cloud management. The different factor that develop are :- ***Software packing:*** The output from software packing will be used on a daily basis during the installation and configuration of various software packages*;* ***Incident management:*** It is used to track any interruptions of issues t the asset management service;

Pool management: It works with AM to make sure that products requested are available on the requested date and duration; ***Release management:*** It controls scheduling and testing of additions and updates on environments; ***Configuration management:*** it helps in the absence of a process with its own repository for assets.

Cloud Governance

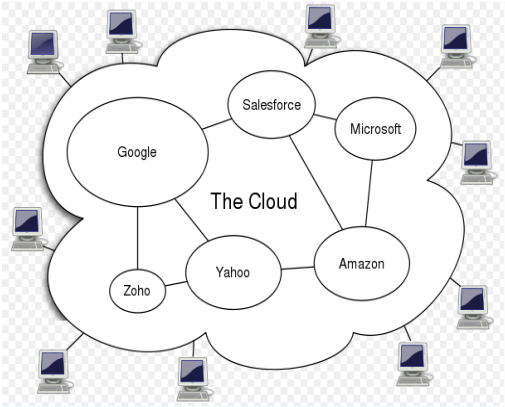
It is the proper definition of roles and responsibilities within an organizational structure. They can be Regulation of new service of creation; Getting more reuses of services; Enforcing standards and best practice; Service change management and service version control.

High Availability (HA) and Disaster Recovery (DR)

HA and DRare some of the important factors for cloud deployment. As cloud is based on service models so different SLAs govern the service based models to avail the service. There are different teams to work on HS and DR : **Mean Time Between Failures (MTBF):** The mean (average) time between successive failures of a given component, subsystem or system; **Mean Time To Recover (MTTR):** The average time that it takes to recover a component, subsystem; **High Availability (HA):** The characteristic of a system that delivers an acceptable or agreed upon high level of service to end users during scheduled periods; **Continuous operations:**The characteristic of asystem that allows an end user to access the system at any time of the day; **Continuous Availability:**Delivering acceptable or agreed to high level of service; **Availability management:**Process of managing IT resources to ensure committed levels of service are achieved to meet upon need of business.Disaster recovery is the process of creating verifying and maintaining on IT continuity plan that is to be executed to restore service in event of disaster :-Protect and maintain currency of vital records; Select site of Vendor that is capable of supporting the requirement of the critical application workload; Provide provision for restoration of all IT services.

Cloud Chargeback Models

Chargeback is a mechanism to institute a free for cloud service type of model. Chargeback allows for IT custodial to position their cloud service are a value added service and use cost recovery mechanism to provide varying degrees of cloud service levels at differentiating costs.The different models available are : **Standard subscription based model:** This type of model entails the total operational cost of IT organization by the total number of applications hosted by the environment.This type of cost recovery is simple to calculate and due to its appeal of simplicity, it finds way in many organizations; **Pay per use model:** This model is targeted for environments with line of business (LOB) of varying sizes and unlike the standard subscription model, this model emphasizes on charging based on applications consumption of resources.This model ensures fair and equitable cost recovery, it may take longer to arrive at agreeable metrics and cost models associated with resource consumption; **Premium pricing model :** It focuses on class of service and generated availability of resources for business applications.LOB's will incur a premium for preferential treatment of application requests, priority in resource allocation, during time of contention to meet service goals fully; **Hybrid model :** The hybrid model attempts to adopt of set breed models and offers the combined advantages of two or more charge back models. This way the flat fee can be characterized as fixed response to be hosted and the additional costs for resource consumption can be linked to variable cost.



Open source software platforms for private clouds, Cloud storage diversity and vendor lock-in and ecological impact

Private clouds provide a cost effective alternative for very large organizations. A private cloud has essentially the same structural components as a commercial one: the servers, the network, Virtual Machines Monitors (VMM) running on individual systems, an archive containing disk images of Virtual Machines (VMs), a front end for communication with theuser, and a cloud control infrastructure. Open-source cloud computing platforms such as *Eucalyptus*, *OpenNebula*, and *Nimbus* can be used as a control infrastructure for aprivate cloud.

Schematically, a cloud infrastructure carries out the following steps to run an application:

- retrieves the user input from the front-end;
- retrieves the disk image of a VM (Virtual Machine) from a repository;
- locates a system and requests the VMM (Virtual Machine Monitor) running on thatsystem to setup a VM;
- invokes the DHCP20 and the IP bridging software to set up a MAC and IP addressfor the VM.

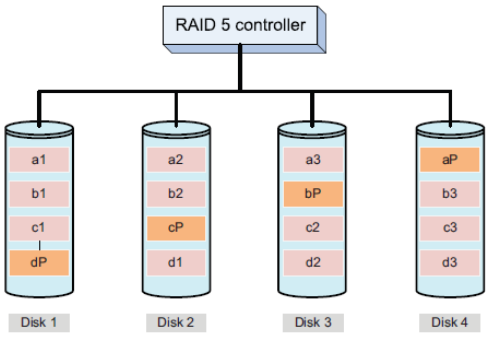
A side-by-side comparison of *Eucalyptus*, *OpenNebula*, and *Nimbus*.

	Eucalyptus	*OpenNebula*	*Nimbus*
Design	Emulate EC2	Customizable	Based on Globus
Cloud type	Private	Private	Public/Private
User population	Large	Small	Large
Applications	All	All	Scientific
Customizability	Administrators limited users	Administrators and users	All but image storage and credentials
Internal security	Strict	Loose	Strict
User access	User credentials	User credentials	x509 credentials
Network access	To cluster controller	-	To each compute node

Cloud storage diversity and vendor lock-in

There are several risks involved when a large organization relies solely on a single cloud provider. As the short history of cloud computing shows, cloud services may be unavailablefor a short, or even for an extended period of time. Such an interruption of service is likely to impact negatively the organization and possibly diminish, or cancel completely, the benefitsof utility computing for that organization. Last, but not least, the single vendor may decide to increase the prices for service andcharge more for computing cycles, memory, storage space, and network bandwidth thanother cloud service providers. The alternative in this case is switching to another provider. Unfortunately, this solution could be very costly due to the large volume of data to be transferred from the old to the new provider. Transferring terabytes or possibly petabytesof data over the network takes a fairly long time and incurs substantial charges for the network bandwidth. A solution to guarding against the problems posed by the vendor lock-up is to replicate the data to multiple cloud service providers. The straightforward replication is verycostly and, at the same time, poses technical challenges. The overhead to maintain data consistency could drastically affect the performance of the virtual storage system consisting of multiple full replicas of the organization's data spread over multiple vendors. Another solution could be based on an extension of the design principle of a RAID-5 system used for reliable data storage.

RAID-5, configuration where individual blocks are stripped overthree disks and a parity block is added; the parity block is constructed by XOR-ing the datablocks, e.g., $aP = a1 \text{ XOR } a2 \text{ XOR } a3$. The parity blocks are distributed among the four disks, *aP* is on disk 4, *bP* on disk 3, *cP* on disk 2, and *dP* on disk 1.



Ecological Impact

The energy consumption of large-scale data centers and their costs for energy and for cooling are significant now and are expected to increase substantially in the future. The same study reports that if the energy demand for bandwidth is 4 Watts-hour perMegabyte32 and if the demand for network bandwidth is 3.2 Gbytes/day/person or 2570Exabytes/year for the entire world population, then the energy required for this activitywill be 1 175 GW. These estimates do not count very high bandwidth applications that mayemerge in the future, such as 3D-TV, personalized immersive

entertainment, such as SecondLife, or massively multi-player online games. The power consumption required by different types of human activities is partially responsiblefor the greenhouse gas emissions. According to a recent study , the greenhousegas emission due to the data centers is estimated to increase from 116 × 106 tonnes of *CO*2in 2007 to 257 tonnes in 2020 due primarily to increased consumer demand. Environmentally Opportunistic Computing is a macro-scale computing idea that exploits the physical and temporal mobility of modern computer processes. A prototype called a Green Cloud is also developed.

Service level agreements, User experience and Software licensing

A Service Level Agreement (SLA) is a negotiated contract between two parties, the customerand the service provider; the agreement can be legally binding or informal and specifies theservices that the customer receives, rather than how the service provider delivers the services. The objectives of the agreement are:

- Identify and define the customers' needs and constraints including the level of resources, security, timing, and quality of service.
- Provide a framework for understanding; a critical aspect of this framework is a clear definition of classes of service and the costs.
- Simplify complex issues; for example, clarify the boundaries between the responsibilities of the clients and those of the provider of service in case of failures.
- Reduce areas of conflict.
- Encourage dialog in the event of disputes.
- Eliminate unrealistic expectations.

An SLA records a common understanding in several areas:

(i) services,

(ii) priorities,

(iii) responsibilities,

(iv) guarantees, and

(v) warranties.

An agreement usually covers: servicesto be delivered, performance, tracking and reporting, problem management, legal compliance and resolution of disputes, customer duties and responsibilities, security, handling ofconfidential information, and termination.

The security threats perceived by this group of users are:

(i) abuse and villainous useof the cloud;

(ii) APIs that are not fully secure;

(iii) malicious insiders;

(iv) account hijacking;

(v) data leaks;

(vi) issues related to shared resources.

A broad set of concerns identified by the NIST working group on cloud security includes:

- Potential loss of control/ownership of data.
- Data integration, privacy enforcement, data encryption.
- Data remanence after de-provisioning.
- Multi tenant data isolation.
- Data location requirements within national borders.
- Hypervisor security.
- Audit data integrity protection.
- Verification of subscriber policies through provider controls.
- Certification/Accreditation requirements for a given cloud service.

Software licensing

Software licensing for cloud computing is an enduring problem without a universally accepted solution at this time. The license management technology is based on the old model of computing centers with licenses given on the basis of named users or as a site license; this licensing technology developed for a centrally managed environment cannot accommodate the distributed service infrastructure of cloud computing or of Grid computing.

A commercial product based on the ideas developed by this research project is *elasticLM* which provides license and billing Web-based services. The architecture of the *elasticLM* license service has several layers: coallocation, authentication, administration, management, business, and persistency. The authentication layer authenticates communications between the license service and the billing service as well as the individual applications; the persistence layer stores the usage records; the main responsibility of the business layer is to provide the licensing service with the licenses prices; the management coordinates different components of the automated billing service.

Challenges of Cloud Computing

The development of efficient cloud applications inherits the challenges posed by the natural imbalance between computing, I/O, and communication bandwidths of physical systems. One of the main advantages of cloud computing, the shared infrastructure, could also have a negative impact. Performance isolation 33 is nearly impossible to reach in a real system, especially when the system is heavily loaded. The performance of virtual machines fluctuates based on the load, the infrastructure services, the environment including the other users. Security isolation is also challenging on multi-tenant systems. Reliability is also a major concern; node failures are to be expected whenever a large number of nodes cooperate for the computations. Choosing an optimal instance (in terms of performance isolation, reliability, and security) from those offered by the cloud infrastructure is another critical factor to be considered. Of course, cost considerations also play a role in the choice of the instance type. Data storage plays a critical role in the performance of any data-intensive application; the organization of the storage, the storage location, as well as, the storage bandwidth must be carefully analyzed to lead to optimal application performance. Clouds support many storage options to set up a file system similar to the Hadoop file system, among them are off-instance cloud storage (e.g. S3), mountable off-instance block storage (e.g., EBS), as well as, storage persistent for the lifetime of the instance.

Architectural styles for cloud computing

Cloud computing is based on the client-server paradigm . The vast majority of cloud applications take advantage of request-response communication between clients and stateless servers. A stateless server does not require a client to first establish a connection to the server, instead it views a client request as an independent transaction and responds to it. The advantages of stateless servers are obvious. Recovering from a server failure requires a considerable overhead for a server which maintains the state of all its connections, while in case of a stateless server a client is not affected while a server goes down and then comes back up between two consecutive requests. For example, a basic web server is stateless; it responds to an HTTP request without maintaining a history of past interactions with the client. The client, a browser, is also stateless since it sends requests and waits for responses.

Several other considerations must be analyzed before deciding on the architectural style of an application. The term neutrality refers to the ability of the application protocol to use different transport protocols such as TCP or UDP, and, in general to run on top of a different protocol stack. For example, we shall see that SOAP can use TCP but also TCP, SMTP35, or JMS36 as transport vehicles. Extensibility refers to the ability to incorporate additional functions such as security. Independence refers to the ability to accommodate different programming styles.

The Common Object Request Broker Architecture (CORBA) was developed in the early 1990s to allow networked applications developed in different programming languages and running on systems with different architecture and system software to work with one another. At the heart of the system is the Interface Definition Language (IDL) used to specify the interface of an object.

Coordination of multiple activities, coordination based on a state machine model

Many cloud applications require the completion of multiple interdependent tasks; the description of a complex activity involving such an ensemble of tasks is known as a workflow. Workflow models are abstractions revealing the most important properties of the entities participating in a workflow management system. *Task* is the central concept in workflow modeling; a task is a unit of work to be performed on the cloud and it is characterized by several attributes, such as

- Name - a string of characters uniquely identifying the task.
- Description - a natural language description of the task.
- Actions - an action is a modification of the environment caused by the execution of the task.
- Preconditions – boolean expressions that must be true before the action(s) of the taskcan take place.
- Postconditions - boolean expressions that must be true after the action(s) of the task do take place.
- Attributes - provide indications of the type and quantity of resources necessary for the execution of the task, the actors in charge of the tasks, the security requirements,whether the task is reversible or not, and other task characteristics.
- Exceptions - provide information on how to handle abnormal events. The exceptions supported by a task consist of a list of <event, action> pairs.

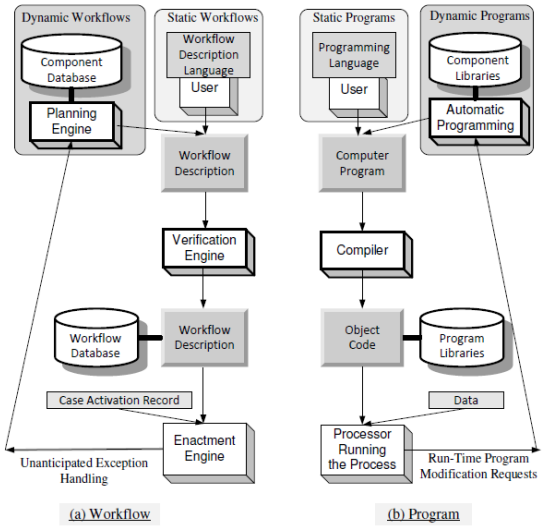
A *composite task* is a structure describing a subset of tasks and the order of their execution.A *primitive task* is one that cannot be decomposed into simpler tasks

A *routing task* is a special-purpose task connecting two tasks in a workflow description.The task that has just completed execution is called the *predecessor* task, the one to be initiated next is called the *successor task*.

A *fork routing task* triggers execution of several successor tasks. Several semantics for this construct are possible:

- All successor tasks are enabled;
- Each successor task is associated with a condition; the conditions for all tasks are evaluated and only the tasks with a true condition are enabled;
- Each successor task is associated with a condition; the conditions for all tasks are evaluated but, the conditions are mutually exclusive and only one conditionmay be true thus, only one task is enabled;
- Nondeterministic, k out of $n > k$ successors are selected at random to be enabled.

A ***join routing task*** waits for completion of its predecessor tasks.



A parallel between workflows and programs.

(a) The life cycle of a workflow.

(b) The life cycle of a computer program.

The workflow definition is analogous to writing aprogram. Planning is analogous to automatic program generation. Verification corresponds to syntactic verification of a program. Workflow enactment mirrors the execution of aprogram. A static workflow corresponds to a static program and a dynamic workflow to a dynamic program.

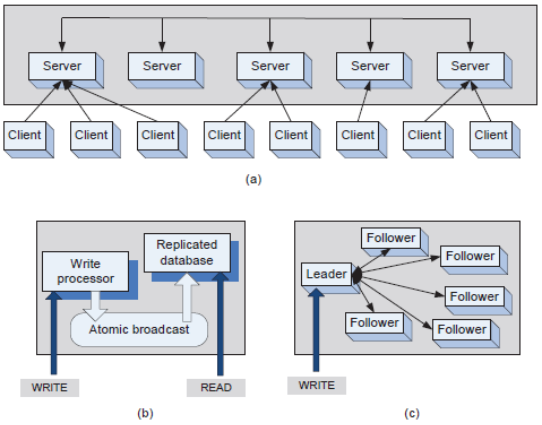
Coordination based on a state machine model

Cloud computing elasticity requires the ability to distribute computations and data across multiple systems; coordination among these systems is one of the critical functions to be exercised in a distributed environment.

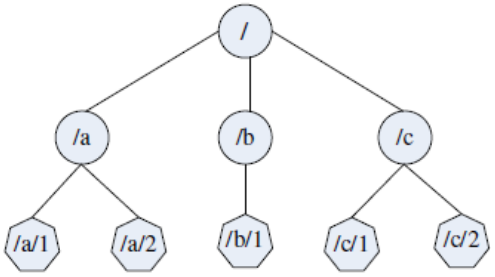
In the distributed data store model the access to data is mitigated by a proxy. This proxy is a single point of failure thus, an architecture with multiple proxies is desirable. The solution for the proxy coordination problem is to consider a proxy as a deterministicfinite state machine which performs the commands sent by clients in some sequence.

ZooKeeper is a distributed coordination service based on this model. The high throughput and low latency service is used for coordination in large-scale distributed systems. The open-source software is written in Java and has bindings for Java and C.

A client uses TCP to connect to a single server; through the TCP connection a client sends requests, receives responses and watch events. A client synchronizes its clock withthe server. If the server fails, the TCP connections of all clients connected to it time-outand the clients detect the failure of the server and connect to other servers.



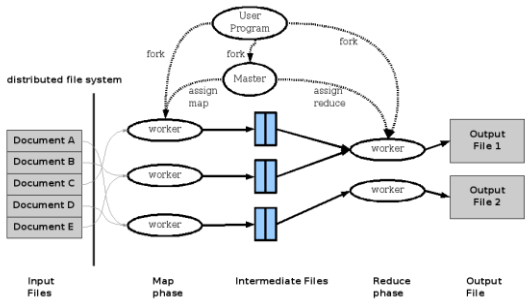
The *ZooKeeper* coordination service. (a) The service provides a single systemimage, clients can connect to any server in the pack. (b) Functional model of the *ZooKeeper* service; the replicated database is accessed directly by *READ* commands, while *WRITE*commands involve a more intricate processing based on atomic broadcast. (c) Processing a*WRITE* command: (1) a server receiving the command from a client, forwards the commandto the *leader*; (2) the *leader* uses atomic broadcast to reach consensus among all *followers*.



ZooKeeper is organized as a shared hierarchical namespace; a name is asequence of path elements separated by a backslash.

The *ZooKeeper* service guarantees:

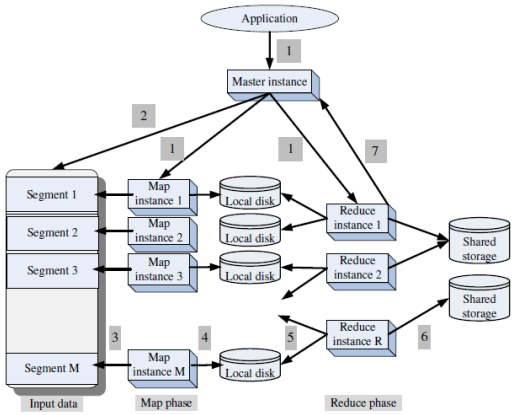
1. Atomicity - a transaction either completes or fails.
2. Sequential consistency of updates - updates are applied strictly in the order they arereceived.
3. Single system image for the clients - a client receives the same response regardless ofthe server it connects.
4. Persistence of updates - once applied, an update persists until it is overwritten by aclient.
5. Reliability - the system is guaranteed to function correctly as long as the majority of servers function correctly.



MapReduce is a programming model inspired by the *map* and the *reduce* primitives of the Lisp programming language. It was conceived for processing and generating large data setson computing clusters.As a result of the computation, a set of input < *key*, *value* >pairs is transformed into a set of output < *key*, *value* >pairs.

Call *M* and *R* the number of *Map* and *Reduce* tasks, respectively, and *N* the number of systems used by the *MapReduce*. When a user program invokes the *MapReduce function*,the following sequence of actions take place:

1. The run-time library splits the input files into *M splits* of 16 to 64 MB each, identifiesa number *N* of systems to run, and starts multiple copies of the program, one of the system being a *master* and the others *workers.* The master assigns to each idle system either a *map* or a *reduce* task. The master makes *O*(*M* +*R*) scheduling decisions andkeeps *O*(*M* × *R*) worker state vectors in memory. These considerations limit the size of *M* and *R*; at the same time, efficiency considerations require that *M*,*R* >> *N*.



The MapReduce philosophy

(1) An application starts a *master instance* and*M* worker instances for the *Map* phase and later *R* worker instances for the *Reduce* phase.

(2) The master partitions the input data in *M* segments.

(3) Each *map instance* reads itsinput data segment and processers the data.

(4) The results of the processing are stored on the local disks of the servers where the *map instances* run.

(5) When all *map instances* have finished processing their data the *R reduce instances* read the results of the first phase and merges the partial results.

(6) The final results are written by the *reduce instances* to a shared storage server.

(7) The *master instance* monitors the *reduce instances* and when all of them report task completion the application is terminated.

2. A worker being assigned a *Map* task reads the corresponding input split, parses< *key*, *value* >pairs and passes each pair to a user-defined *Map* function. The intermediate< *key*, *value* >pairs produced by the *Map* function are buffered in memorybefore being written to a local disk, partitioned into *R* regions by the partitioning function.
3. The locations of these buffered pairs on the local disk are passed back to the master,who is responsible for forwarding these locations to the reduce workers. A reduce worker uses remote procedure calls to read the buffered data from the local disks ofthe map workers; after reading all the intermediate data, it sorts it by the intermediate keys. For each unique intermediate key, the key and the corresponding set of intermediate values are passed to a user-defined *Reduce*

function. The output of the *Reduce* function is appended to a final output file.

4. When all *Map* and *Reduce* tasks have been completed, the master wakes up the user program.

Cloud for Science and Engineering

The generic problems in virtually all areas of science are:

- Collection of experimental data.
- Management of a very large volumes of data.
- Building and execution of models.
- Integration of data and literature.
- Documentation of the experiments.
- Sharing the data with others; data preservation for a long periods of time.

All these activities require powerful computing systems.

A typical example of a problem faced by agencies and research groups is data discovery in large scientific data sets. Examples of such large collections are the biomedical and genomicdata at NCBI, the astrophysics data from NASA, or the atmospheric data from NOAA and NCAR. The process of online data discovery can be viewed as an ensemble of several phases : (i) recognition of the information problem; (ii) generation of search queries using one or more search engines; (iii) evaluation of the search results; (iv) evaluation of the web documents; and (v) comparing information from different sources. The web search technology allows the scientists to discover text documents related to such data but, the binary encoding of many of them poses serious challenges.

High performance computing on a cloud

Some of the codes used are:

Community Atmosphere Mode **(CAM)**the atmospheric component of CCSM (CommunityClimate System Model) is used for weather and climate modeling45. The code developed at NCAR uses two two-dimensional domain decompositions; one for the dynamics and the other for re-mapping.

General Atomic and Molecular Electronic Structure System **(GAMESS)** is used for ab-initioquantum chemistry calculations.

Gyrokinetic (GTC): is a code for fusion research48. It is a self-consistent, gyrokinetic tridimensional Particle-in-cell (PIC) code with a non-spectral Poisson solver.

Integrated Map and Particle Accelerator Tracking Time (IMPACT-T) is a code for the prediction and performance enhancement of accelerators

MAESTRO is a low Mach number hydrodynamics code for simulating astrophysical flows.Its integration scheme is embedded in an adaptive mesh refinement algorithm based on a hierarchical system of rectangular non-overlapping grid patches at multiple levels with different solution.

MImd Lattice Computation **(MILC)** is a QCD (Quantum Chromo Dynamics) code used to study "strong" interactions binding quarks into protons and neutrons and holding them together in the nucleus.

The authors of HPCC (High Performance Computing Challenge) benchmark to compare the performance of *EC2* with the performance of three large systems atNERSC. HPCC (High Performance Computing Challenge) is a suite of seven synthetic benchmarks: three targeted synthetic benchmarks which quantify basic system parameters that characterize individually the computation and communication performance; four complex synthetic benchmarks which combine computation and communication and can beconsidered simple proxy applications.

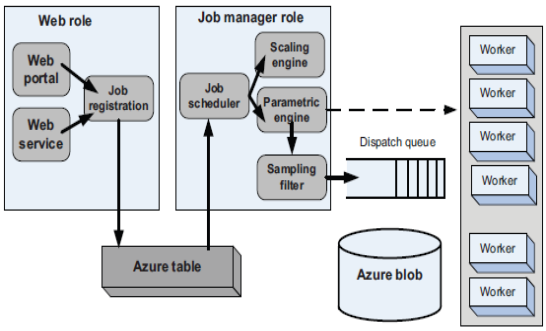
Cloud computing for biology research

Biology is one of the scientific fields that needs vast amounts of computing power and was one of the first to take advantage of cloud computing. Molecular dynamics computations are CPU-intensive while protein alignment is data-intensive.

Azure offers VM with four levels of computing power depending on the number of cores:small (1 core), medium (2 cores), large (8 cores), and extra large (>8 cores). The experimentused 8 core CPUs with 14 GB RAM and a 2 TB local disk. It was estimated that the computation would take six to seven CPU-years thus, the experiment was allocated3,700 weighted instances or 475 extra-large VMs from three data centers; each data center hosted three *AzureBLAST* deployments, each with 62 extra large instances. The 10 million sequences were divided into multiple segments, each segment was submitted for execution by one *AzureBLAST* deployment. With this vast amount of resources

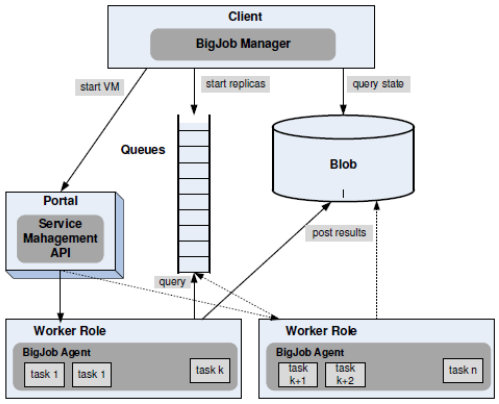
allocated, it took14 days to complete the computations which produced 260 GB of compressed data spreadacross over 400 000 output files.

Each task is associated with a record in the task table and this state record is updated periodically by the worker instance running the task; the progress of the task is monitored by the manager. The dispatch queue feeds into a set of worker instances. A worker periodically updates the task state in the task table and listens for any control signals from the manager.



Cirrus, a general platform for executing legacy Windows applications on the cloud.

Figure illustrates the use of a software system called *BigJob* to decouple resource allocation from resource binding for the execution of loosely coupled workloads on an Azure platform. this software eliminates the need for the application to manage individualVMs. The results of measurements show a noticeable overhead for starting VMs and forlaunching the execution of an application task on a remote resource; increasing the computing power of the VM decreases the completion time for long-running tasks.



The execution of loosely-coupled workloads using the Azure platform.

Social computing

Social networks play an increasingly important role in people's lives; they have expanded in terms of the size of the population involved and in terms of the function performed.A promising solution for analyzing large-scale social networks data is to distribute the computation workload over a large number of nodes of a cloud. Traditionally, the importance of a node or a relationship in a network is done using sampling and surveying but, in a very large network structural properties cannot be inferred by scaling up the results from small networks; it turns out that the evaluation of social closeness is computationally intensive.

Social intelligence is another area where social and cloud computing intersect. Indeed,the process of knowledge discovery and techniques based on pattern recognition demand high performance computing and resources that can be provided by computing clouds.Case-based reasoning (CBR), the process of solving new problems based on the solutions of similar past problems, is used by context-aware recommendation systems; it requiressimilarity-based retrieval. As the case base accumulates, such applications must handle massive amount of history data and this can be done by developing new reasoning platforms running on the cloud. CBR is preferable to rule-based recommendation systems for large scale social intelligence applications. Indeed, the rules can be difficult to generalize or applyto some domains, all triggering conditions must be strictly satisfied, scalability is a challengeas data accumulate, and the systems are hard to maintain as new rules have to be added as the amount of data increases.

Relation between digital content and cloud computing

A system based on CBR consists of a cloud layer, a case-based reasoning engine, and an API. The cloud layer uses the Hadoop Distributed File System clusters to store application data represented by cases, as well as, social network information, such as relationship topology, and pairwise social closeness information. The CBR engine calculates similarity measures between cases to retrieve the most similar ones and also stores new cases back to the cloud layer; the API connects to a master node which is responsible for handling user queries, distributes the queries to server machines, and receives results.

A case consists of a problem description, solution, and optional annotations about the path to derive the solution. The CBR uses *MapReduce*; all the cases are grouped by their *userId*, and then a *breath first search* (BFS) algorithm is applied to the graph where each node corresponds to one user. *MapReduce* is used to calculate the closeness according to pairwise relationship weight. A reasoning cycle has four steps:

(a) retrieve the most relevant or similar cases from memory to solve the case;

(b) reuse - map the solution from the priorcase to the new problem;

(c) revise - test the new solution in the real world or in a simulation and, if necessary, revise;

(d) retain - if the solution was adapted to the target problem,store the result as a new case.

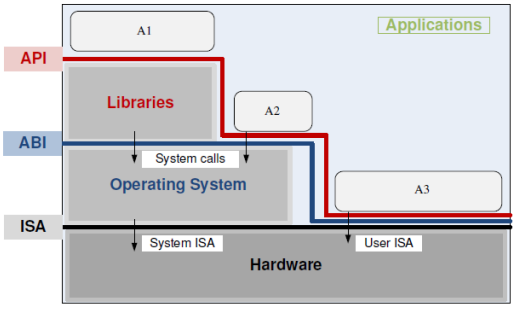
There are numerous examples of monitoring, notification, presence, location, and map services based on the Svc approach including: *Monitor Mail*, *Monitor RSSFeed*, *Send SMS*,*Make Phone Call*, *GTalk*, *Fireeagle*, and *Google Maps*. As an example, consider a service to send a phone call when a specific Email is received; the *Mail Monitor* Svc uses input parameters such as: User Id, Sender Address Filter, Email Subject Filter, to identify an Email and generates an event which triggers the *Make TTS Call* action of a *Text To SpeechCall* Svc linked to it.

The system in supports creation, deployment, activation, execution and management of Event Driven Mashups; it has a user interface, a graphics tool called Service Creation Environment that supports easily the creation of new Mashups, and a platform called *Mashup Container* that manages Mashup deployment and execution. The system consists of two sub-systems, the *service execution platform* for Mashups execution and the *deployer* module that manages the installation of Mashups and Svcs. A new Mashup is created using the graphical development tool and it is saved as an XML file; it can then be deployed into a *Mashup Container* following the Platform as a Service (PaaS) approach. The *Mashup Container* supports a primitive Service Level Agreement allowing the delivery of different levels of service.

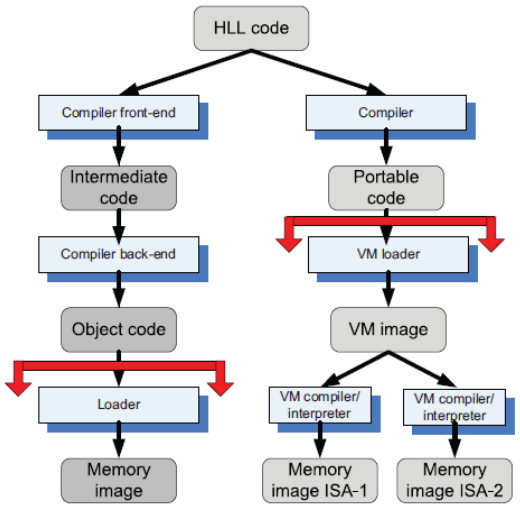
Layering and virtualization

A common approach to manage system complexity is to identify a set of *layers* with well defined *interfaces* among them.

The ISA (Instruction Set Architecture) defines the set of instructions of a processor,for example, the Intel architecture is represented by the *x86*-32 and *x86*-64 instruction sets for systems supporting 32-bit addressing and 64-bit addressing, respectively. The hardware supports two execution modes, a *privileged*, or *kernel* mode and a *user* mode.



Layering and interfaces between layers in a computer system; the software components including applications, libraries, and operating system interact with the hardware via several interfaces: the Application Programming Interface (API), the Application Binary Interface (ABI), and the Instruction Set Architecture (ISA). An application uses library functions (A1), makes system calls (A2), and executes machine instructions (A3).



High Level Language (HLL) code can be translated for a specific architecture and operating system; the HLL code can also be compiled into portable code and then the portable code be translated for systems with different ISAs. The code shared/distributed is the object code in the first case and the portable code in the second case.

Virtualization defined, virtualization benefits, server virtualization, virtualization for x86 architecture.

Virtualization is an abstraction layer that decouples the physical hardware from the operating system to deliver greater IT resources utilization and flexibility virtualization helps in**Save Money**: the number of physical server is reduced. This helps in saving cost; **Dramatically Increase Control**: virtualization provides a flexibility foundation to provide capacity on demand for organization; **Simplify Disaster Recovery**: more efficiency and cost effective disaster recovery solution can be realized; **Business Readiness Assessment**: virtualization introduces shared computing model and there is no need to implement physically.

Virtualization Benefits

The virtualization has many benefits as follows:- Server consideration and reduced hardware cost, Green IT and reduced power of cooling, Increased availability/ business continuity and disaster recovery, Maximized hardware resources and reduced administration and labor costs, Efficient

application and desktop software development and maintenance, Dynamic and extensible infrastructure to rapidly address new business requirements.

Server Virtualization

Server virtualization covers different types of virtualization such as client, storage, network virtualization. Server virtualization is the masking of server resource including the number of identity of individual physical server, processors and spacing system from server users.

Virtualization for the x86 Architecture

Virtualization on processors encounters a set of challenges that the virtualization on RISC processors do not have. Therefore forward and backward compatibilities must be considered when designing virtualization for x86.Most operating system, includes those for x86 such as Windows & Linux are designed to run on basic hardware, so they naturally assume that they fully own the computer hardware. The x86 architecture offers four levels of privilege known as ring 0,1,2 & 3 to OS and mange access to the computer hardware.Virtualizing the x86 architecture requires placing a virtualization layer under the operating system to create and manage the virtual machine that deliver shared resources.

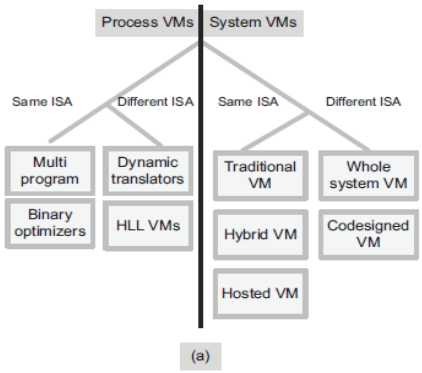
Virtual machine monitors

A *Virtual Machine Monitor (VMM)* also called *hypervisor* is the software that securely partitions the resources of computer system into one or more virtual machines. A *gues toperating system* is an operating system that runs under the control of a VMM rather than directly on the hardware. The VMM runs in kernel mode while a guest OS runs in user mode; sometimes the hardware supports a third mode of execution for the guest OS.The VMMs allow several operating systems to run concurrently on a single hardware platform; at the same time, VMMs enforce isolation among these systems, thus security. A VMM controls how the guest operating system uses the hardware resources; the events occurring in one VM do not affect any other VM running under the same VMM. At thes ame time, the VMM enables:

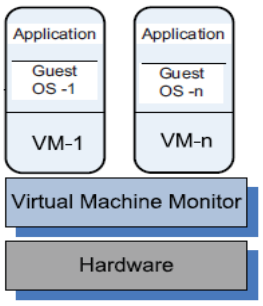
- Multiple services to share the same platform.
- The movement of a server from one platform to another, the so-called live migration.
- System modification while maintaining backward compatibility with the original system.

Virtual machines

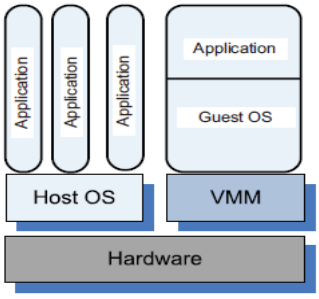
A *Virtual Machine (VM)* is an isolated environment that appears to be a whole computer, but actually only has access to a portion of the computer resources. Each virtual machine appears to be running on the bare hardware, giving the appearance of multiple instances ofthe same computer, though all are supported by a single physical system. Virtual machines have been around since early 1970s when IBM released its VM 370 operating system.



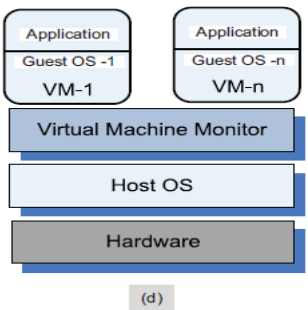
(a)



(b)



(c)



(d)

(a) A taxonomy of process and systems VMs for the same and for different Instruction Set Architectures (ISAs). Traditional, Hybrid, and Hosted are three classes of VMs for systems with the same ISA.

(b) Traditional VMs; the VMM supports multiple virtual machines and runs directly on the hardware;.

(c) Hybrid VM; the VMM shares the hardware with a host operating system and supports multiple virtual machines.

(d) HostedVM; the VMM runs under a host operating system.

Performance and Security Isolation

Performance isolation is a critical condition for Quality of Service (QoS) guarantees in shared computing environments. Indeed, if the run-time behavior of an application is affected by other applications running concurrently and thus, competing for CPU cycles, cache, main memory, disk and network access, it is rather difficult to predict the completion time; moreover, it is equally difficult to optimize the application. Several operating systems including Linux/RK, QLinux , and SILK support some performance isolation, but problems still exist as one has to account for all resources used and has to distribute the overhead for different system activities, including context switching and paging,to individual users, a problem often described as *QoS crosstalk*

Processor virtualization presents multiple copies of the same processor or core on multicore systems; the code is executed directly by the hardware, while *processor emulation* presents a model of another hardware system; instructions are "emulated" in software much slower than virtualization. An example is Microsofts Virtual PC could run on chip sets other than the *x86* family; it was used on Mac hardware until Apple adopted Intel chips.

Traditional operating systems multiplex multiple processes or threads while a virtualization supported by a VMM (Virtual Machine Monitor) multiplexes full operating systems.Obviously, there is a performance penalty as an OS is considerably more heavy weight thana process and the overhead of context switching is larger. A Virtual Machine Monitor executes directly on the hardware a subset of frequently used machine instructions generatedby the application and emulates privileged instructions including device I/O requests. The subset of the instructions executed directly by the hardware includes arithmetic instructions , memory access and branching instructions.

The security vulnerability of VMMs is considerably reduced as the systems expose a much smaller number of privileged functions; for example, the *Xen* VMM can be accessed through 28 hypercalls while a standard Linux allows hundreds.

Full virtualization and paravirtualization

For computer architecture to support virtualization and allow a Virtual Machine Monitor to operate efficiently

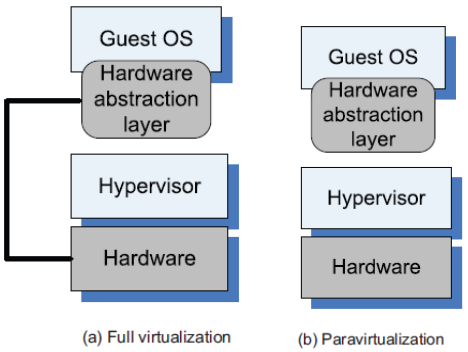
- A program running under the VMM should exhibit a behavior essentially identical tothat demonstrated when running on an equivalent machine directly.
- The VMM should be in complete control of the virtualized resources.
- A statistically significant fraction of machine instructions must be executed without the intervention of the VMM.

Another way to identify an architecture suitable for a virtual machine is to distinguish two classes of machine instructions, sensitive, instructions which require special precautions at execution time and innocuous, instructions that are not sensitive. In turn, sensitive instructions can be:

- *Control sensitive*, instructions that attempt to change either the memory allocation,or the privileged mode.
- *Mode sensitive*, instructions whose behavior is different in the privileged mode.

An equivalent formulation of the conditions for efficient virtualization can be based on this classification of machine instructions: *a VMM for a third or later generation computer can be constructed if the set of sensitive instructions is a subset of the privileged instructions of that machine.* To handle non-virtualizable instructions one could resort to two strategies:

- Binary translation. The VMM monitors the execution of guest operating systems ; non-virtualizable instructions executed by a guest operating system are replaced withother instructions.
- Paravirtualization. The guest operating system is modified to use only instructions that can be virtualized.



(a) Full virtualization (b) Paravirtualization

(a) The full virtualization requires the hardware abstraction layer of the guestOS to have some knowledge about the hardware.

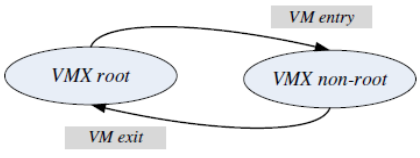
(b) The paravirtualization avoids this requirement and allows full compatibility at the Application Binary Interface (ABI).

Hardware support for virtualization

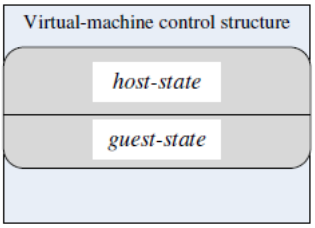
In the early 2000 it became obvious that hardware support for virtualization was necessary and Intel and AMD started work on the first generation virtualization extensions of the*x86*59 architecture. In 2005 Intel released two Pentium 4 model supporting *VT-x* and in2006 AMD announced Pacifica and then several Athlon 64 models.

The various problems faced by virtualization of the *x86*architecture:

- *Ring deprivileging;* this means that a VMMs forces the guest software, the operating system and the applications, to run at a privilege level greater than 0.
- *Ring aliasing*; problems created when a guest OS is forced to run at a privilege level other than that it was originally designed for.
- *Address space compression*
- *Non-faulting access to privileged state*
- *Guest system calls*
- *Interrupt virtualization*
- *Access to hidden state*
- *Ring compression*
- *Frequent access to privileged resources increases VMM overhead*



(a)



(b)

(a) The two modes of operation of *VT-x*, and the two operation to transit fromone to another; (b) The Virtual Machine Control Structure (VMCS) includes *host-state* and*guest-state* areas which control the *VM entry* and *VM exit* transitions.

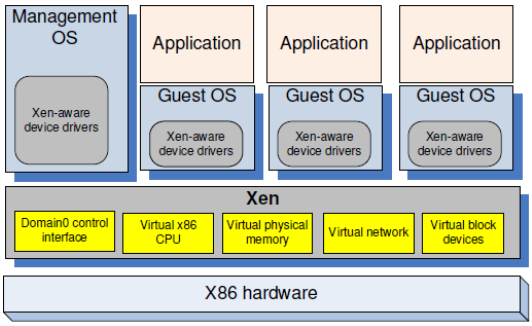
• *VMX root:* intended for VMM operations, and very close to the *x86* without *VT-x*.

• *VMX non-root:* intended to support a VM.

Case study:

Xen is a Virtual Machine Monitor (VMM) or hypervisor developed by the Computing Laboratoryat the University of Cambridge,

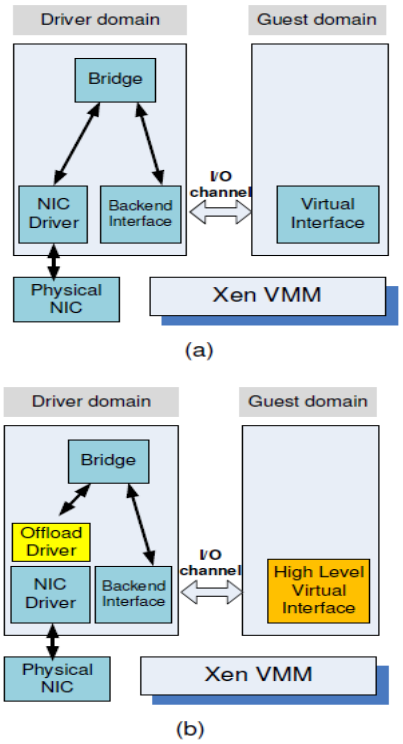
The goal of the Cambridge group led by Ian Pratt was to design a VMM capable ofscaling to about 100 virtual machines running standard applications and services without any modifications to the Application Binary Interface (ABI). Fully aware that the *x86*architecture does not support efficiently full virtualization, the designers of *Xen* opted for paravirtualization.



Xen for the *x86* architecture; in the original *Xen* implementation a guestOS could be either *XenoLinix*, *XenoBSD*, or *XenoXP*. The management OS dedicated tothe execution of *Xen* control functions and privileged instructions resides in *Dom0*; guestoperating systems and application reside in *DomU*.

Optimization of network virtualization in Xen 2.0

A virtual machine monitor introduces a significant network communication overhead. Forexample, it is reported that the CPU utilization of a *VMware Workstation 2.0* systemrunning Linux 2.2.17 was 5−6 times higher than that of the native system (Linux 2.2.17)in saturating a 100 Mbps network .



Xen network architecture (a) The original architecture; (b) The optimized architecture.

In other words, handling the same amount of trafficas the native system to saturate the network, the VMM executes a much larger number ofinstructions, 5 − 6 times larger.

The *Xen* network optimization discussed in [241] covers optimization of:

(i) the virtual interface;

(ii) the I/O channel; and

(iii) the virtual memory.

The effects of these optimizations are significant for the send data rate from the optimized *Xen* guest domain, an increasefrom 750 to 3 310 Mbps and rather modest for the receive data rate, 970 versus 820 Mbps.

The next target of the optimization effort is the communication between the guest domain and the driver domain. Rather than copying a data buffer holding a packet, each packet is allocated in a new page and then the physical page containing the packet is remapped into the target domain; for example, when a packet is received, the physical page is re-mapped to the guest domain.

The third optimization covers the virtual memory. The virtual memory in *Xen* 2.0takes advantage of the *superpage* and *global page mapping* hardware features available onPentium and Pentium Pro processors.

vBlades

The goal of the *vBlades* project was to create a VMM for the *Itanium*family of *IA64* Intel processors70, capable of supporting the execution of multiple operating systems in isolated protection domains with security and privacy enforced by the hardware.*Itanium* was selected because of its multiple functional units and multi-threading support. The hardware has an *IVA register* to maintain the address of the *Interruption VectorTable*; the entries in this table control both the interrupt delivery and the interrupt state collection. Different types of interrupts is not disabled. Each guest OS maintains its own version of this vector table and has its own IVA register; the hypervisor uses the guest OS IVA register to give control to the guest interrupt handler when an interrupt occurs.

A number of *privileged-sensitive* instructions behave differently function of the privilege level. The VMM replaces each one of them with a privileged instruction during the dynamic transformation of the instruction stream. Among the instructions in this category are:

- cover, saves stack information into a privileged register; the VMM replaces it with a break.b instruction.
- thash and ttag, access data from privileged virtual memory control structures and have two registers as arguments.

Access to performance data from performance data registers is controlled by a bit in the *Processor Status Register* with the PSR.sp instruction.

Memory virtualization is guided by the realization that a VMM should not be involved in most of memory read and write operations to prevent a significant degradation of the performance, but, at the same time, the VMM should exercise a tight control and prevent a guest OS to act maliciously.

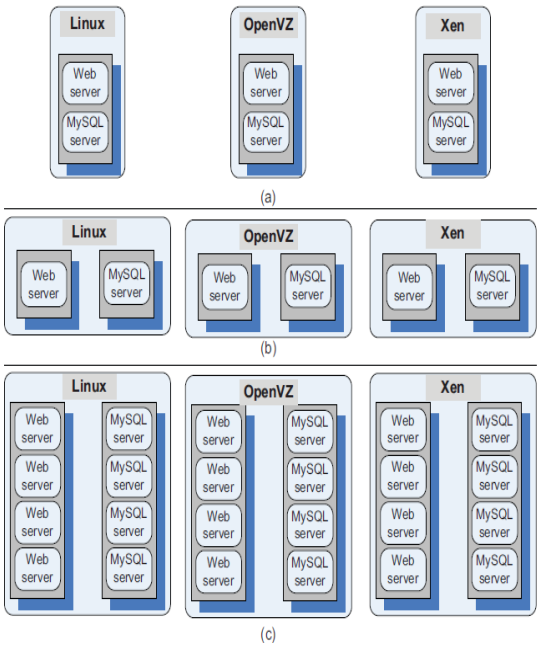
A performance comparison of virtual machines

The memory allocation in *OpenVZ* is more flexible than in the case of paravirtualization memory not used in one virtual environment can be used by others. The system uses a common file system; each virtual environment is a directory of files isolated using chroot.To start a new virtual machine one needs to copy the files from one directory to another,create a config file for the virtual machine, and launch the VM.

OpenVZ has a two level scheduler: at the first level, the fair-share scheduler allocatesCPU time slices to containers based on cpuunits values; the second level is a standard Linux scheduler which decides what process to run in that container. The I/O scheduler isalso two-level; each container has an I/O priority, and the scheduler distributes the available I/O bandwidth according to the priorities.

The general question is whether consolidation of the applications and servers is a good strategy for cloud computing; the specificquestions examined are:

- How the performance scales up with the load?
- What is the impact of a mix of applications?
- What are the implications of the load assignment on individual servers?



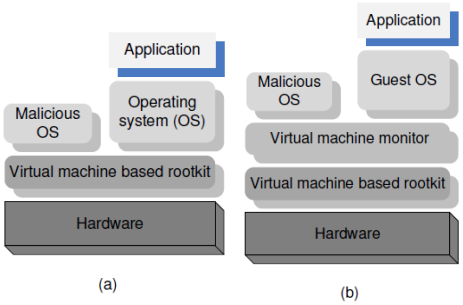
The setup for the performance comparison of a native Linux system with *OpenVZ*,and the *Xen* systems. The applications are a web server and a MySQL database server.

(a)The first experiment, the web and the DB, share a single system;

(b) The second experiment,the web and the DB, run on two different systems;

(c) The third experiment, the web andthe DB, run on two different systems and each has four instances.

The darker side of virtualization



The insertion of a *Virtual-Machine Based Rootkit* (VMBR) as the lowest layer of the software stack running on the physical hardware;

(a) below an operating system;

(b) below a legitimate virtual machine monitor. The VMBR enables a malicious OS to run surreptitiously and makes it invisible to the genuine or the guest OS and to the application. It is also feasible to insert the VMBK between the physical hardware and a "legitimateVMM." As a virtual machine running under a legitimate VMM sees a virtual hardware, theguest OS will not notice any change of the environment; so the only trick is to present the legitimate VMM with a hardware abstraction, rather than allow it to run on the physical hardware.

Hypervisor Management Software

For each hypervisor, there is a companion layer of Hypervisor Management Software that provides range of function such like Create VM, Delete VM, Move VM etc as the hypervisor management controlling function the hypervisor. A unique set of API's and GUI's is available for each Hypervisior/ Hypervisor management software pair that is used by the client IT staff.Hypervisor is the foundation for virtualization on server, enabling hardware to be divided into multiple logical partitions and ensuring isolation among them. This also supports Ethernet transport mechanism and Ethernet switch which are needed for VLAN capability. VLAN allows secure communication between logical partitions without using any physical Ethernet Adaptor.

Logical Partitioning

Logical Partitioningis a subset of computer's hardware resources, virtualized as a separate computer. In effect, a physical machine can be partitioned into multiple logical partitions, each hosting a separate operating system. Logical partitioning divides hardware resources. Two LPARs may access memory from a common memory chip, provided that the ranges of addresses directly accessible to each do not overlap.LPARs safely allow combining multiple test, development, quality assurance, and production work on the same server.

Virtual IO server (VIO)

VIO is a technique used in enterprise environments to lower costs, improve performance and makes server management easy and simple. The virtual I/O methodology allows a single physical adapter card to be seen as multiple virtual network interface cards (NICs) and virtual host bus adapters.Some of the important features of virtual I/O include: Cost Savings: Virtual I/O helps to lower costs by allowing easy and simple server management;

Fewer Cables: Virtual I/O requires only a single cable to connect servers to both storage and the network. Multiple I/O cables are thus replaced with a single cable.; Increased I/O Density: Virtual I/O technique increases the I/O density by enabling more connections; Simplified Management: Virtual I/O enables greater use of virtual NICs and HBAs while providing greater flexibility.

Virtual infrastructure requirements

Virtualization products have strict requirements on backend infrastructure components including storage, backup, system management, security and Time Sync. Ensuring that these components are of required configuration is critical for successful implementation.What ever applicable, enterprise tools are used to gain a clear understanding of the environment and the configuration and utilization of various system. A virtualization sizing tool is that used to accurately calculate the size of a potential virtualization platform.

Storage Virtualization

Storage Virtualization improves the utilization of storage and people assets because it allows one to treat resource as single pool, accessing and managing those resources across your organization more efficiently.

Benefits

Storage is made simpler, Make storage more heterogeneous, Make storage more manageable.Cloud can access, design, develop,optimize and support on demand, infrastructures that are integrated, virtualized and autonomic and built on open standards.

Storage Area Networks

Storage Area Network is a method of provisioning by locally attaching the device to the operating system, to the servers with the SAN architecture. We can connect different type of disk array depends on the storage devices.Network attached storage is different with respect to SAN as it uses the file based protocols such as NFS. In this architecture it is evident that the storage is available remotely and can be accessed as a file and not as a disk block. SANS are becoming a passive technology.

Cloud server virtualization

Virtual Machine technology enables customer to run multiple operating systems concurrently in a single physical server. Virtual infrastructure server solution addresses a set of key customer scenarios including consolidating and automating software testing and development environment, migrating legacy application consolidating multiple server workloads and testing distributed server applications on a single physical server.

Virtualized Data Center

Virtual data centers provide flexibility of hardware by abstracting the memory, processor, storage etc available in the form of virtual resources. Virtualized data centers provide all management and control functions of the environment under the same umbrella. They are: It uses a very a very simple provisioning method to allocate the virtual servers with easy to use VI; Business needs special attention at different intervals, virtual servers help to automate the operational needs of the deployment; It helps to schedule and alert the different management, controls and support functions with scheduled automated tasks; It is a very good tool for making the utilization of process memory and IOPS requirements with detailed report; It helps to set the customized roles based on the type of work as well as access permission to resources.

Policies and mechanisms for resource management

A policy typically refers to the principles guiding decisions, while mechanisms represent the means to implement policies. Separation of policies from mechanisms is a guiding principle in computer science.

Cloud resource management policies can be loosely grouped into five classes:

1. Admission control.

2. Capacity allocation.

3. Load balancing.

4. Energy optimization.

5. Quality of service (QoS) guarantees.

The explicit goal of an admission control policy is to prevent the system from accepting workload in violation of high-level system policies; for example, a system may not accept additional workload which would prevent it from completing work already in progress or contracted. Load balancing and energy optimization can be done locally, but global load balancing and energy optimization policies encounter the same difficulties as the one. Virtually all optimal, or near-optimal, mechanisms to address the five classes of policies do not scale up and typically target a single aspect of resource management e.g., admission control, but ignore energy conservation.

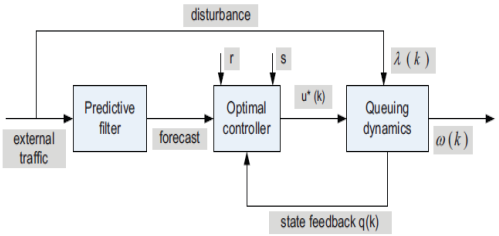
The four basic mechanisms for the implementation of resource management policies are:

- Control theory.
- Machine learning.
- Utility-based.

- Market-oriented/economic mechanisms.

Applications of control theory to task scheduling on a cloud

Control theory has been used to design adaptive resource management for many classes of applications including: power management, task scheduling, QoS adaptation in web servers, and load balancing. The classical feedback control methods are used in all these cases to regulate the key operating parameters of the system based on measurement of the system output; the feedback control in these methods assumes a linear time-invariant system model, and a closed-loop controller. This controller is based on an open-loop system transfer function which satisfies stability and sensitivity constraints.



The structure of an optimal controller described a above ; the controller uses the feedback regarding the current state as well as the estimation of the future disturbance due to environment to compute the optimal inputs over a finite horizon. The two parameters *r* and *s* are the weighting factors of the performance index.

The discrete-time optimal control problem is to determine the sequenceof control variables $u(i)$, $u(i + 1), \ldots, u(n-1)$ to minimize the expression

$$J(i) = \Phi(n, x(n)) + \sum_{k=i}^{n-1} L^k(x(k), u(k))$$

where $\Phi(n, x(n))$ is the cost function of the final step, n, and $Lk(x(k), u(k))$ is a timevarying cost function at the intermediate step k over the horizon $[i, n]$. The minimization is subject to the constraints.

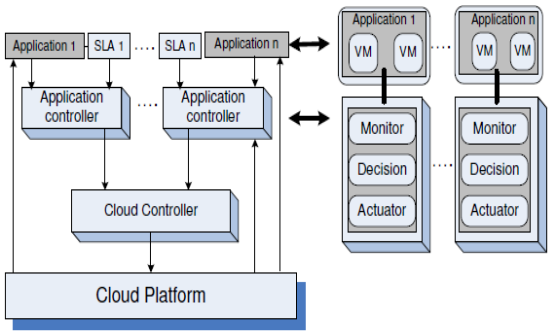
$$x(k+1) = f^k(x(k), u(k)).$$

where $x(k+1)$, the system state at time $k+1$, is a function of $x(k)$, the state at time k, and of $u(k)$, the input at time k; in general, the function fk is time-varying thus, its superscript.

Stability of a two-level resource allocation architecture

The main components of a control system are: the inputs, the control system components, and the outputs. The inputs in such models are: the offered workload and the policiesfor admission control, the capacity allocation, the load balancing, the energy optimization, and the QoS guarantees in the cloud.

The system components are *sensors* used to estimate relevant measures of performance and *controllers* which implement various policies; the output is the resource allocations to the individual applications.



A two-level control architecture; application controllers and cloud controllers work in concert.

There are three main sources of instability in any control system:

[1]. The delay in getting the system reaction after a control action;

[2]. The granularity of the control, the fact that a small change enacted by the controllersleads to very large changes of the output;

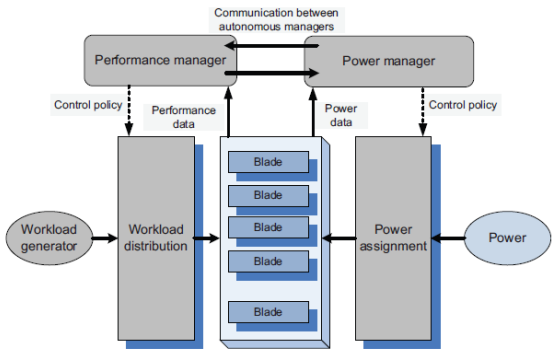
[3]. Oscillations, when the changes of the input are too large and the control is too weak, such that the changes of the input propagate directly to the output.

Feedback control based on dynamic thresholds

The elements involved in a control system are sensors, monitors, and actuators. The *sensors*measure the parameter(s) of interest, then transmit the measured values to a *monitor* whichdetermines if the system behavior must be changed, and, if so, it requests the *actuators* tocarry out the necessary actions. Often, the parameter used for admission control policy is the current system load; when a threshold, e.g., 80%, is reached the cloud stops accepting additional load. In practice the implementation of such a policy is challenging, or outright infeasible.First, due to the very large number of servers and to the fact that the load changes rapidly in time, the estimation of the current system load is likely to be inaccurate. Second, the ratio of average to maximal resource requirements of individual users specified in a service level agreement is typically very high; once an agreement is in place user demands must be satisfied; user requests for additional resources within the SLA limits cannot be denied.

Coordination of specialized autonomic performance managers

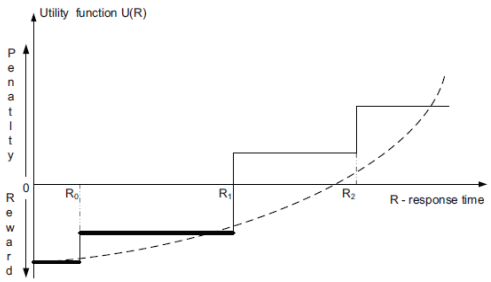
Virtually all modern processors support DVS (Dynamic Voltage Scaling) as a mechanism for energy saving; indeed, the energy dissipation scales quadratically with the supply voltage.The power management controls the CPU frequency thus, the rate of instruction execution; for some compute-intensive workloads the performance decreases linearly with the CPUclock frequency, while for others the effect of lower clock frequency is less noticeable or non existent. The clock frequency of individual blades/servers is controlled by a power manager typically implemented in the firmware; it adjusts the clock frequency several times a second.



Autonomous performance and power managers cooperate to ensure SLA prescribed performance and energy optimization; they are fed with performance and power data and implement the performance and power management policies, respectively.

Give Utility based model for cloud-based Web services

A *utility function* relates the "benefits" of an activity or service with the "cost" to providethe service. For example, the benefit could be revenue and the cost could be the power consumption. A service level agreement (SLA) often specifies the rewards as well as penalties associated with specific performance metrics. Sometimes the quality of services translates into averageresponse time; this is the case of cloud-based web services when the SLA often specifies explicitly this requirement.



The utility function $U(R)$ is a series of step functions with jumps corresponding to the response time, $R = R0|R1|R2$, when the reward and the penalty levels change accordingto the SLA. The dotted line shows a quadratic approximation of the utility function.

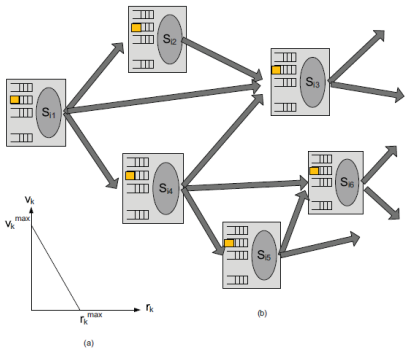


Figure shows the case when the performance metrics is R, the response time. The largest reward can be obtained when $R \leq R_0$; a slightly lower reward corresponds to $R_0 < R \leq R_1$

(a) The utility function, *vk* the revenue (or the penalty) associated with a response time *rk* for a request of class $k \in K$; the slope of the utility function is $mk = -vmaxk/rmaxk$.

(b) A network of multiqueues; at each server *Si* there are $|K|$ queues for each one of the$k \in K$ classes of requests. A tier consists of all requests of class $k \in K$ at all servers $Sij \in I, 1 \leq j \leq 6$.

Resource bundling and Combinatorial auctions for cloud resources

Resources in a cloud are allocated in *bundles*; users get maximum benefit from a specific combination of resources. Indeed, along with CPU cycles, an application needs specific amounts of main memory, disk space, network bandwidth, and so on. Resource bundling complicates traditional resource allocation models and has generated an interest in economic models and, in particular, in auction algorithms. In the context of cloud computing, anauction is the allocation of resources to the highest bidder.

Pricing and allocation algorithms

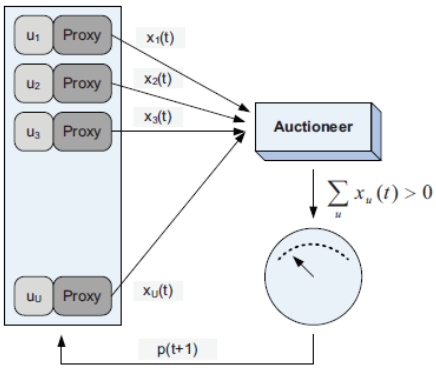
A pricing and allocation algorithm partitions the set of users in two disjoint sets, winners and losers, denoted as *W* and *L*, respectively;the algorithm should:

- Be computationally tractable; traditional combinatorial auction algorithms such asVickey-Clarke-Groves (VLG) fail this criteria, they are not computationally tractable.
- Scale well; given the scale of the system and the number of requests for service, scalability is a necessary condition.

- Be objective; partitioning in winners and losers should only be based on the price $\pi u$of a user's bid; if the price exceeds the threshold then the user is a winner, otherwise the user is a loser.
- Be fair; make sure that the prices are *uniform*, all winners within a given resource pool pay the same price.
- Indicate clearly at the end of the auction the unit prices for each resource pool.
- Indicate clearly to all participants the relationship between the supply and the demand in the system.

The function to be maximized is

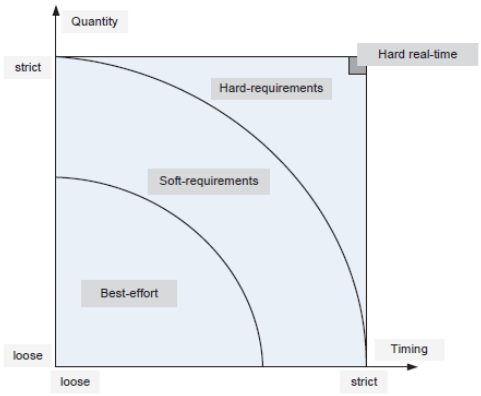
$$\max_{x,p} \; f(x,p).$$



The schematics of the ASCA algorithm; to allow for a single round auction users are represented by proxies which place the bids *xu*(*t*). The auctioneer determines if there is an excess demand and, in that case, it raises the price of resources for which the demand exceeds the supply and requests new bids.

Scheduling algorithms for computing clouds

Scheduling is a critical component of the cloud resource management. Scheduling is responsiblefor resource sharing/multiplexing at several levels; a server can be shared among several virtual machines, each virtual machine could support several applications, and each application may consist of multiple threads. CPU scheduling supports the virtualization of a processor, the individual threads acting as virtual processors; a communication link canbe multiplexed among a number of virtual channels, one for each flow.



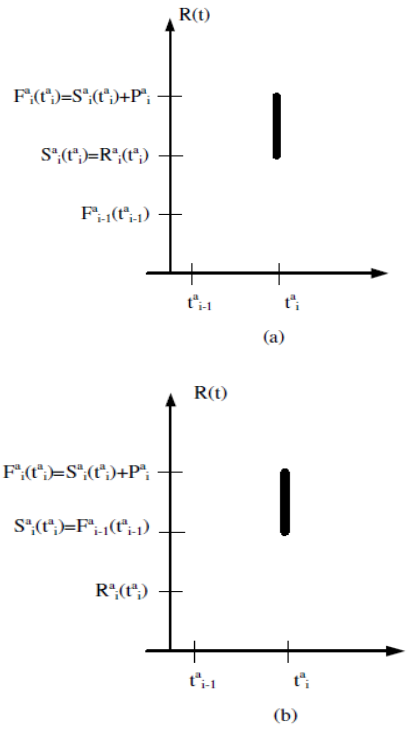
Best-effort policies do not impose requirements regarding either the amount of resources allocated to an application, or the timing when an application is scheduled.Soft-requirements allocation policies require statistically guaranteed amounts and timing constraints while hard-requirements allocation policies demand strict timing and precise amounts of resources.

Fair queuing

Computing and communication on a cloud are intimately related, therefore, it should be no surprise that the first algorithm we discuss can be used for scheduling packets transmission,as well as threads. Interconnection networks allow cloud servers to communicate with one another and with the users; these networks consist of communication links of limited bandwidth and switches/routers/gateways of limited capacity. When the load exceedsits capacity, a switch starts dropping packets because it has limited input buffers for the switching fabric and for the outgoing links, as well as, limited CPU cycles. First,it introduces a *bit-by-bit round-robin (BR)* strategy; as the name implies, in this rather impractical scheme a single bit from each queue is transmitted and the queues are visited in a round-robin fashion. Let *R*(*t*) be the number of rounds of the BR algorithm up to time*t* and *Nactive*(*t*) the number of active flows through the switch. Call *tai*the time when the packet *i* of flow *a*, of size *Pai* bits arrives and call *Sai* and *Fai* the values of *R*(*t*) when thefirst and the last bit, respectively, of the packet *i* of flow *a* are transmitted.

Then,

$$F_i^a = S_i^a + P_i^a \quad \text{and} \quad S_i^a = \max\left[F_{i-1}^a, R(t_i^a)\right].$$



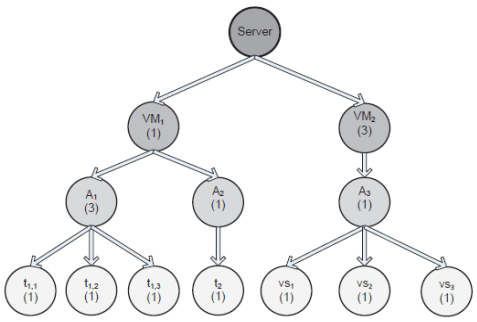
Transmission of a packet i of flow a arriving at time t^a_i of size P^a_i bits. The transmission starts at time $S^a_i = \max[F^a_{i-1}, R(t^a_i)]$ and ends at time $F^a_i = S^a_i + P^a_i$ with $R(t)$ the number of rounds of the algorithm. (a) The case $F^a_{i-1} < R(t^a_i)$. (b) The case $F^a_{i-1} \geq R(t^a_i)$.

Start-time fair queuing

A hierarchical CPU scheduler for multimedia operating systems was proposed in. The basic idea of the *start-time fair queuing (SFQ)* algorithm is to organize the consumers ofthe CPU bandwidth in a tree structure; the root node is the processor and the leaves of thistree are the threads of each application. A scheduler acts at each level of the hierarchy; the fraction of the processor bandwidth, *B*, allocated to the intermediate node *i* is

$$\frac{B_i}{B} = \frac{w_i}{\sum_{j=1}^{n} w_j}$$

with *wj* , $1 \leq j \leq n$, the weight of the *n* children of node *i*



The SFQ tree for scheduling when two virtual machines *VM*1 and *VM*2 run on a powerful server. *VM*1 runs two best-effort applications *A*1, with three threads *t*1;1, *t*1;2, and*t*1;3, and *A*2 with a single thread *t*2; *VM*2 runs a video-streaming application *A*3 with threethreads *vs*1, *vs*2, and *vs*3.

Cloud scheduling subject to deadlines

Often, a Service Level Agreement specifies the time when the results of computations done on the cloud should be available. This motivates us to examine cloud scheduling subject to deadlines, a topic drawing from a vast body of literature devoted to real-time applications.

Task characterization and deadlines. Real-time applications involve periodic or a periodic tasks with deadlines;

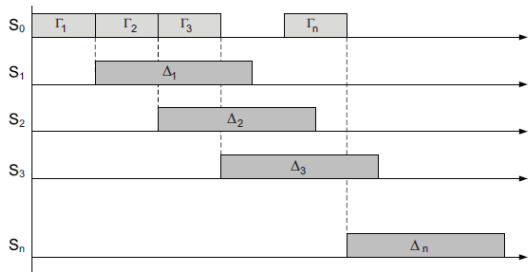
System model. The application runs on a partition of a cloud, a virtual cloud witha *head node* called *S*0 and *n worker nodes S*1, *S*2, . . . , *Sn*. The system is homogeneous, all workers are identical, and the communication time from the head node to any worker nodeis the same. The head node distributes the workload to worker nodes and this distribution is done sequentially. In this context there are two important problems:

1. The order of execution of the tasks Π*i*.

2. The workload partitioning and the task mapping to worker nodes.

Scheduling policies The most common scheduling policies used to determine theorder of execution of the tasks are:

- FIFO - First-In-First-Out - the tasks are scheduled for execution in the order of theirarrival.
- EDF - Earliest Deadline First, the task with the earliest deadline is scheduled first.
- MWF - Maximum Workload Derivative First.



The timing diagram for the Equal Partitioning Rule; the algorithm assigns an equal workload to individual worker nodes , $\alpha i = 1/n$. The head node, *S*0, distributes sequentially the data to individual worker nodes

Scheduling** MapReduce **applications subject to deadlines

Several options for scheduling Apache *Hadoop*, an open source implementation of the *MapReduce* algorithm are:

1. The default FIFO schedular.
2. The Fair Scheduler [381].
3. The Capacity Scheduler.
4. The Dynamic Proportional Scheduler.

The two assumptions for our initial derivation:

- The system is homogeneous, ρm and ρr, the cost of processing a unit data by the *map*and the *reduce* task, respectively, are the same for all servers.
- Load equipartition.

Resource management and dynamic application scaling

The demand for computing resources such as CPU cycles, primary and secondary storage,and network bandwidth depend heavily on the volume of data processed by an application.The demand for resources can be a function of the time of the day, can monotonically increase or decrease in time, or can experience predictable, or unpredictable peaks. For example, a new web service will experience a low request rate at the very beginning andthe load will exponentially increase if the service is successful. A service for income tax processing will experience a peak around the tax filling deadline, while access to a service provided by FEMA (Federal Emergency Management Agency) will increase dramatically after a natural disaster.

Cloud security risks

There are multiple ways to look at the security risks for cloud computing.

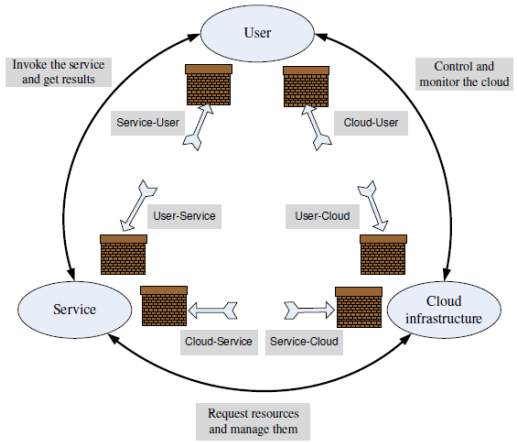
i. traditional security threats,
ii. threats related to system availability,
iii. threats related to third-party data control.

Traditional threats are those experienced for some time by any system connected to the Internet, but with some cloud-specific twists.

It begins at the user site; the user must protect the infrastructure used to connect to the cloud and to interact with the application running on the cloud. This task is more difficult because some components of this infrastructure are outside the firewall protecting the user.

Availability of cloud services is another major concern. System failures, power outages, and other catastrophic events could shutdown cloud services for extended periods of time; when such an event occurs data lock-in could prevent a large organization whose business model depends on these data to function properly.

Third-party control generates a spectrum of concerns caused by the lack of transparency and limited user control.



Surfaces of attacks in a cloud computing environment

Security: the top concern for cloud users

Virtually all surveys report that security is the top concern for cloud users who are accustomed to have full control of all systems where sensitive information is stored and processed.Users typically operate inside a secure perimeter protected by a corporate firewall. In spiteof the potential threats, users have to extend their trust to the cloud service provider if they wish to benefit from the economical advantages of utility computing. This is a fairly difficult transition, yet a critical one for the future of cloud computing. To support this transition some argue that cloud security is in the hands of experts, hence users are evenbetter protected than when they are in charge with their own security.Major user concerns are the unauthorized access to confidential information and the data theft. Data is more vulnerable in storage, than while it is being processed. Data is kept in storage for extended periods of time, while it exposed to threats during processing for relatively short time. Hence, close attention should be paid to the security of storage servers and to data in transit.

Privacy: privacy impact assessment

The term privacy refers to the right of an individual, a group of individuals, or an organizationto keep information of personal nature or proprietary information from being disclosed. Many nations view privacy as a basic human right. Consumers online would be required to comply with the four widely-accepted fair information practices.

i. Notice

ii. Choice

iii. Access

iv. Security

Operating system security

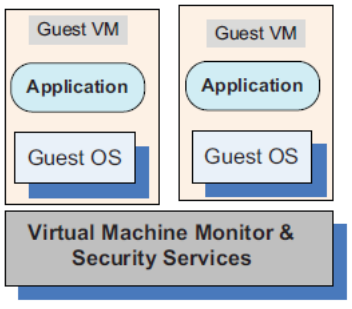
An operating system (OS) allows multiple applications to share the hardware resources of a physical system subject to a set of policies. A critical function of an OS is to protect applications against a wide range of malicious attacks such as unauthorized access to privileged information, tempering with executable code, and spoofing. Such attacks can now target even single-user systems such as personal computers, tablets, or smart phones. Data brought in the system may contained malicious code; this could be the case of a Java applet, or of data imported by a browser from a malicious web site.

The *mandatory security* of an OS is considered to be: "any security policy where the definition of the policy logic and the assignment of security attributes is tightly controlled by a system security policy administrator." Access control, authentication usage, and cryptographic usage policies are all elements of the mandatory OS security.

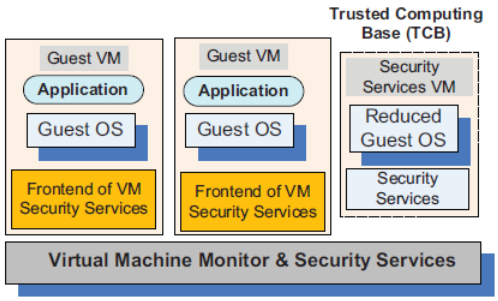
Virtual machine security

The hybrid and the hosted VM models , expose the entire system to the vulnerability of the host operating system thus, we will not analyze these models. Virtual security services are typically provided by the VMM. Another alternative is to have a dedicated security services VM. Another alternate is to have a dedicated security service VM.

A secureTCB (Trusted Computing Base) is a necessary condition for security in a virtual machine environment; if the TCB is compromised then the security of the entire system is affected.



(a)



(b)

(a) Virtual security services provided by the VMM; (b) A dedicated securityVM.

A guest OS runs on simulated hardware and the VMM has access to the state of all virtual machines operating on the same hardware. The state of a guest virtual machine can be saved, restored, cloned, and encrypted by the VMM.

Security of virtualization

There are several useful implications of this fact:

1. Ability to support the Infrastructure as a Service (IaaS) delivery model; in this model a user selects an image matching the local environment used by the application and then uploads and runs the application on the cloud using this image.
2. Increased reliability; an operating system with all the applications running under it can be replicated and switched to a hot standby102 in case of a system failure.
3. Straightforward mechanisms to implement resource management policies:
   - To balance the load of a system, an OS and the applications running under it can be moved to another server when the load on the current server exceeds a high water mark.
   - To reduce power consumption the load of lightly loaded servers can be moved to other servers and then turn off or set on standby mode the lightly loaded servers.

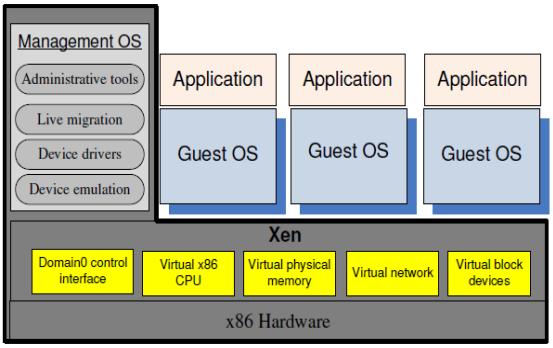
Security risks posed by shared images

Even when we assume that a cloud service provider is trustworthy there are other sources of concern many users either ignore, or underestimate the danger they pose. One of them, especially critical for the Infrastructure as a Service (IaaS) cloud delivery model, is image sharing. For example, a user of AWS has the option to choose between Amazon Machine Images (AMIs) accessible through the *Quick Start* or the *Community AMI* menus of the *EC2* service. The option of using one of these AMIs is especially tempting for a first time user, or for a less sophisticated one.

Two procedures for the creation of an image are available, ec2-bundle-image and ec2-bundle-volume. The first is used for images prepared as loopback files105 when the data is transferred to the image in blocks. To bundle a running system the creator of the image can use the second procedure when bundling works at the level of the file system and files are copied recursively to the image.

TheSecurity risks posed by a management OS

A hypervisor supports a stronger isolation between the VMs running under it than the isolation between processes supported by a traditional operating system. Yet the hypervisor must rely on a management OS to create VMs and to transfer data in and out from a guest VM to storage devices and network interfaces.



The trusted computing base of a *Xen*-based environment includes the hardware, *Xen*, and the management operating system running in *Dom0*. The management OS supports administrative tools, live migration, device drivers, and device emulators.

A trusted virtual machine monitor

- The TVMM should support not only traditional operating systems, by exporting the hardware abstraction for open-box platforms, but also the abstractions for closed box platforms
- An application should be allowed to build its software stack based on its needs. Applications requiring a very high level of security, e.g., financial applications and electronic voting systems should run under a very thin OS supporting only the functionality required by the application and the ability to boot.
- Support additional capabilities to enhance system assurance:
  - Provide trusted paths from a user to an application.
  - Support attestation,
  - Provide air-tight isolation guarantees for the TVMM by denying the platform administrator the root access.

Amazon Web Services

The services are grouped in several categories:

a. Computing and networking,

b. storage and content delivery,

c. deployment and management,

d. databases,

e. application services.

To access AWS one must first create an account at *http://aws.amazon.com/*; once the account is created the Amazon Management Console (AMC) allows the user to select one of the service, e.g., *EC2* and then start an instance.

The next step is to create an AMI (Amazon Machine Image) on one of the platforms supported by AWS and start an instance using the *RunInstance* API; if the application needs more than 20 instances then a special form must be filled out. The local instance store persists only for the duration of an instance; the data will persist if an instance is started using the Amazon EBS (Elastic Block Storage) and then the instance can be restarted at a later time.

The *Network & Security* panel allows the creation of *Security Groups*, *Elastic IP addresses*, *Placement Groups*, *Load Balancers* and *Key Pairs* while the EBSpanel allows the specification of volumes and the creation of snapshots.

Connecting clients to cloud instances through _rewalls

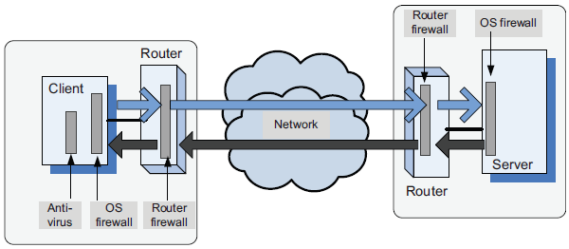
A firewall is a software system based on a set of rules for filtering network traffic.

A *rule* specifies a filtering option at:

a. the network layer, when filtering is based on the destination/source IP address;

b. the transport layer, when filtering is based on destination/ source port number;

c. the MAC layer, when filtering is based on the destination/ source MAC address.



Firewalls screen incoming and sometimes outgoing traffic. The first obstacle encountered by the inbound or outbound traffic is a router firewall, the next one is the firewall provided by the host operating system; sometimes, the antivirus software provides a third line of defense.

***Security rules for application- and transport-layer protocols in** EC2*

A virtual machine running under Amazon's *EC2* has several IP addresses:

1. *EC2 Private IP Address:*
2. *EC2 Public IP Address:*
3. *EC2 Elastic IP Address:*

Amazon Web Services use *security groups* to control access to user's virtual machines. A virtual machine instance belongs to one, and only one, security group, which can only be defined before the instance is launched. Once an instance is running, the security group the instance belongs to cannot be changed. However, more than one instance can belong to a single security group.

The following steps allow the user to add a security rule:

1. Sign in to the AWS Management Console at *http://aws.amazon.com* using your Email address and password and select *EC2* service.
2. Use the *EC2 Request Instance Wizard* to specify the instance type, whether it should be monitored, and specify a key/value pair for the instance to help organize and search,
3. Provide a name for the key pair, then on the left hand side panel choose *Security Groups* under *Network & Security*, select the desired security group

EC2 Request Instance Wizard is used to:

a. specify the number and type of instances and the zone;

b. specify the kernelId, the RAM diskId, and enable the *Cloud-Watch* service to monitor the *EC2* instance;

c. add tags to the instance; a tag is stored in the cloud and consists of a case-sensitive key/value pair private to the account.

How to launch an** EC2 **Linux instance and connect to it

1. Launch an instance
2. Connect to the instance using *ssh* and the *tcp* transport protocol

sudo iptables -A iptables -p -tcp -dport ssh -j ACCEPT

Enter the Linux command

ssh -i abc.pem ec2-user@PublicDNSname

3. Gain root access to the instance

sudo –i

4. Run the service ServiceName

nohup ServiceName

nohup ServiceName > p.out 2 > p.err &

How to use** S3 **in Java

The Java API for Amazon Web Services is provided by the AWS SDK

Create an *S3* client. *S3* access is handled by the class *AmazonS3Client* instantiated with the account credentials of the AWS user.

```
AmazonS3Client s3 = new AmazonS3Client(
new BasicAWSCredentials("your_access_key",
"your_secret_key"));
```

The access and the secret keys can be found on the user's AWS account home page Buckets. An *S3 bucket* is analogous to a file folder or directory and it is used to store *S3Objects*. Bucket names must be *globally unique* hence, it is advisable to check first if thename exists.

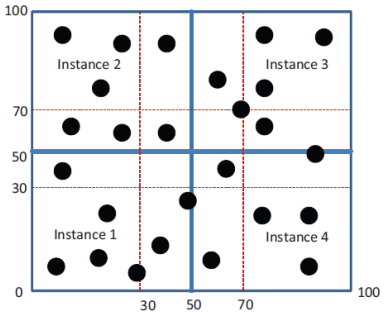
```
s3.doesBucketExist("bucket_name");
File f = new File("local_file_name");
s3.putObject("bucket_name", "key", f);
S3Object myFile = s3.getObject("bucket_name",
"key");
InputStream in = myFile.getObjectContent();
ObjectListing listing =
s3.listObjects("bucket_name");
List<S3ObjectSummary> summaries =
listing.getObjectSummaries();
```

Cloud-based simulation of a distributed trust algorithm

This expectation is motivated by several reasons:

- The convenience of data access from any site connected to the Internet.
- The data transfer rates of wireless networks are increasing; the time to transfer data to and from a cloud is no longer a limiting factor.
- The mobile devices have limited resources; while new generations of smart phones and tablet computers are likely to use multi-core processors and have a fair amount of memory, power consumption is and will continue to be a major concern in the near future. Thus, it seems reasonable to delegate compute-

intensive and data-intensive tasks to an external entity, e.g., a cloud.



Data partitioning for the simulation of a trust algorithm; the area covered is of size 100 × 100 units. The nodes in the four sub-areas of size 70 × 70 units are processed by an instance of the cloud application.

A cloud service for adaptive data streaming

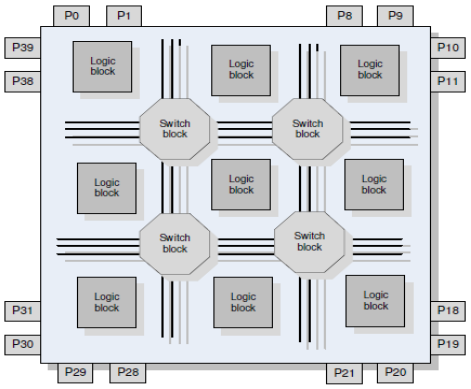
Data streaming is the name given to the transfer of data at a high-rate with real-time constraints. Multimedia applications such as music and video streaming, high-definition television (HDTV),scientific applications which process a continuous stream of data collected by sensors, the continuous backup copying to a storage medium of the data flow within a computer, andmany other applications require the transfer of real-time data at a high-rate. For example,to support real-time human perception of the data, multi-media applications have to makesure that enough data is being continuously received without any noticeable time lag.

Another design decision is how the two services should interact to optimize the performance; two alternatives come to mind:

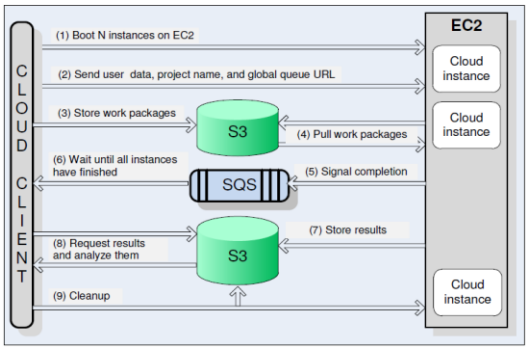
1. The audio service running on the *EC2* platform requests the data file from the *S3*, converts it, and, eventually, sends it back. The solution involves multiple delays and it is far from optimal.
2. Mount the *S3 bucket* as an *EC2* drive. This solution reduces considerably the start-up time for audio streaming.

Cloud-based optimal FPGA synthesis

A cloud is an ideal running environment for scientific applications which involve model development; in this case multiple cloud instances could concurrently run slightly different models of the system. When the model is described by a set of parameters, the application can be based on the SPMD (Same Program Multiple Data) paradigm combined with an analysis phase when the results from the multiple instances are ranked based on a well-defined metric.



The structure of an FPGA with 30 pins, *P*1-*P*29, 9 logic blocks and 4 switchblocks.



The architecture of a cloud-based system to optimize the routing and placement of components on an FPGA.

SOA Journey to Infrastructure

The path to transformation consists of a long journey with staged approach, leading to ultimate goal of service-oriented enterprise, multiple islands of disparate infrastructure in today's environment need to be consolidated to gain controls, reduced costs and becomes operationally efficient.The next step is to introduce virtualized infrastructure to improve utilization levels and allow dynamic flexibility to move resources and capacity to meet fluctuating workload demands.

SOA and Cloud

SOA serves as the foundation for the move into cloud computing and it owns the characteristics if a cloud including a shared infrastructure, self service capabilities and the fact that it will be virtualized.SOA infrastructure is required in order to effectively apply service orientation to an enterprise or large scale software component development, basically SOA infrastructure consists of middleware, physical infrastructure and management all together covering non-functional requirements.

SOA Defined

SOA is an approach to architecture that is intended to promote flexibility through encapsulation and loose coupling, SOA functionality are defined and

exposed as 'service' and there is only one instant of each service implementation.SOA is defined by what a service is services are defined by the following characteristics:- Explicit, implementation – independent interfaces, Loosely bound, Invoked through communication protocols, Stress location transparency and interoperability and Encapsulate reusable business function.

SOA and IAAS

The IT analysts recommend an IT infrastructure that is:- Shared across customers, units and applications; Dynamically driven by business policies and service level requirement.IT virtualization is viewed as a technological aspect of cloud infrastructure in order to create a pool of infrastructure resources such as computing power data storage in order to mask the physical nature of boundaries from users. Cloud infrastructure has many service components, however they need not all be implemented concurrently. Services can be divided into four domains, application services, information services, common IT services, infrastructure services. Within each domain, SOA can be measured and chartered across a continuum of increasing dynamism.

SOA based cloud Infrastructure

Organizations intent upon leverage cloud infrastructure should consider the following steps :- **Analysis and strategy:-**It is recommended to have an incremental, phased approach for adopting SOA and cloud infrastructure. A good starting point is to conduct a business innovation assessment; Planning:-Once the business needs and IT gaps are identified, a strategic and tactile planning effort can be launched to develop on IT value case and road map for incremental IT transformation; Implementation:-It is an excellent idea to establish an enterprise level SOA as well as the development and run time environment standards before the first set of SOA projects are launched; Value-driven:-It is important to note the purpose of SOA and cloud infrastructure are to improve the business performance, flexibility, agility so that IT can support business at business speed.

ACKOWLEDGEMENTS

The Author thanks all who directly or indirectly contributed for developing this article purely with Academic interest , so that it will be convenient for the students to develop themselves with basic knowledge in Cloud Computing and thereby motivate them for further reading .

REFERENCES

[1]. Dan C Marinescu: Cloud Computing Theory and Practice. Elsevier(MK) 2013.

[2]. Rajkumar Buyya , James Broberg, Andrzej Goscinski: Cloud Computing Principles and Paradigms, Willey 2014.

[3]. John W Rittinghouse, James F Ransome:Cloud Computing Implementation, Management and Security, CRC Press 2013

[4]. Kumar Saurabh, Cloud Computing, Wiley India,2011.

[5]. Michael Miller, Cloud Computing: Web based applications that change the way you work and collaborate online, Que publishing , August 2009

[6]. Haley Beard, Cloud Computing Best Practices for Managing and Measuring Processes for On Demand computing applications and data Centers in the Cloud with SLAs, Emereo Pty Limited, July 2008

[7]. Millard, Christopher (2013). Cloud Computing Law. Oxford University Press. ISBN 978-0-19-967168-7.

[8]. Singh, Jatinder; Powles, Julia; Pasquier, Thomas; Bacon, Jean (July 2015). "Data Flow Management and Compliance in Cloud Computing". IEEE Cloud Computing **2** (4): 24–32. doi:10.1109/MCC.2015.69.

[9]. Armbrust, Michael; Stoica, Ion; Zaharia, Matei; Fox, Armando; Griffith, Rean; Joseph, Anthony D.; Katz, Randy; Konwinski, Andy; Lee, Gunho; Patterson, David; Rabkin, Ariel (1 April 2010). "A view of cloud computing". Communications of the ACM **53** (4): 50. doi:10.1145/1721654.1721672.

[10]. Hu, Tung-Hui (2015). A Prehistory of the Cloud. MIT Press. ISBN 978-0-262-02951-3.

[11]. Mell, P. (2011, September 31). The NIST Definition of Cloud Computing. Retrieved November 1, 2015, from National Institute of Standards and Technology website: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

